ST MAGAZINE PRESENTE

STMAGICIEL

HS N°0001

Le premier hors-série de ST Magazine à parution aléatoire et au contenu plus que discutable

23 Juillet 1991 - Canada : 5,50 $C - Belgique : 145 FB - Suisse : 6 FS - Bretagne : 20 francs.

20 F

LA MICRO, C'EST LA MORT !

Alors comme ça, on fait les difficiles, on trouve que le ST, c'est plus assez bien pour Môssieur? C'est la fin des haricots, le ST est foutu, il va falloir commencer à faire des stocks de trucs et de machins? Avant de le condamner, visez un peu comment ça se passait, avant, il y a, pffioouuu, bien longtemps. Ca nous rajeunit pas, ces conneries.

GENERATION

C'est pas pour faire prétentieux, mais merde, les petits jeunes, ils ne connaissent rien à rien. Parce qu'ils ont tâté du ST et de l'Amiga, ils ont tout vu, ils sont sûrs de pouvoir faire une carrière d'informaticien. Laissez-moi doucement rigoler. Petits cons.

TOUT EST CHAOS

Je dis "petits cons", j'ai mes raisons. Tu leur annonces qu'une machine coûte plus de 5000 francs, et ils s'arrachent les cheveux, ils poussent des grands cris, il faut leur mettre des grandes taloches pour les calmer. Au fond, ce que je pense, c'est que Papa Tramiel (Atari) et Tonton Sugar (Amstrad) ont bien réussi: nous voilà avec une génération de nains qui pensent qu'un ordinateur ne doit pas coûter plus de 3000 francs, doit comporter au moins un Mega de RAM, et crachent dessus si le processeur mouline à moins de 8MHz. Je vous disais, des petits cons. Alors que la micro-informatique, mis à part les deux consumeristes mégalos cités plus haut, ça a TOUJOURS été cher, et pas forcément performant pour autant. Des exemples, j'en ai plein, passons au paragraphe suivant, ou bien ça sera le bordel.

TOUS MES IDEAUX

Arbitrairement, je vais commencer par le PET, de Commodore, laissons tomber les KIM-1 et les Proteus III, c'est beaucoup trop galère, ça ne m'intéresse pas. Le PET, à l'époque, à la fin des années 70, c'était un petit bijou: on pouvait toucher avec ses doigts à soi 8 Ko de RAM, on tripotait son premier 6502, on dessinait avec des caractères graphiques, ça donnait du 200 par 40 et on était heureux. Il était pas beau, mais on était heureux, pourvu qu'on allongeât les 7500 balles, ils étaient souriants et polis, à l'époque, les revendeurs. Aujourd'hui, macache, ils te prennent ton chèque de deux barres avec un air forcé, c'est à peine s'ils ne te crachent pas sur les chaussures.

DES MOTS ABIMES

Avec des trucs à 7500 balles, forcément, tu ne touchais pas grand-monde, et les clubs micro-informatiques, tu les comptais sur les doigts de la main, tous dans les beaux quartiers, re-forcément. Les industriels de l'informatique ont réfléchi, ils ont regardé leurs ordinateurs: à qui on pouvait bien les refiler sans tomber dans le gouffre du grand-public, saloperie de grand-public qui ne veut que des prix bas. Ils ont réfléchi encore un peu, il faut ce qu'il faut, et puis on a eu droit à de beaux Goupil dans les écoles, des machines qui plantaient quand ça leur plaisait, mais au moins ça plantait français, et c'était orange, c'était joli. Plus tard, via des plans informatiques extrêmement coûteux et inutiles, on a eu les Silz et les Thomson, qui n'étaient ni oranges ni jolis, mais qui continuaient de planter français, on a sa fierté, oui ou merde?

Le Mur des Lamentations, lieu de rendez-vous de tous les informaticiens, qu'ils s'appellent David ou non.

JE CHERCHE UNE AME

Heureusement, les constructeurs étrangers ne se tournaient pas les pouces, et il commençait à pleuvoir des machines aux carosseries bigarrées, purée que c'était moche. Il y avait eu le TRS 80, un truc au clavier que si tu le touches tu meurs d'apoplexie, mais qui avait un Basic et une résolution graphique qui tenaient la route.

En parlant de résolution, une machine qui eut un beau succès d'estime, ce fut le DAI, une machine qui non contente d'être belge offrait par ailleurs 16 couleurs en 512 par 244. Pour une machine conçue au tout début des années 80, c'était géant, sans parler des sons, sur quatre voix, quand même, le tout piloté par un 8080 et essayable chez soi après avoir lâché une bonne douzaine de Pascal. On en était fous, vous vous rappelez? Oui, je sais, je n'ai pas parlé de l'Apple, c'est voulu. Tout le monde le connait, il n'y a rien à dire, si ce n'est que c'était cher, qu'il y avait plein de softs, qu'il était super, et que je crevais d'envie d'en avoir un, voilà.

QUI POURRA M'AIDER

Voilà pour les éléphants, les dinosaures. Parce qu'après, à partir de 84-85, il est sorti plein de nouvelles machines, un vrai plaisir pour les canards de micro. C'est vrai, rappelez-vous, il en sortait quasiment tous les mois, ça sortait de partout, on nous vendait du test machine à tire-larigot, c'était génial. Les jeunes, aujourd'hui, ils ne connaissent pas ça. L'Atari ST, l'Amiga, et le PC plus le Mac, à la rigueur, et voilà. Je vais encore jouer sur la fibre nostalgique: que celui qui ne s'amusait pas à comparer les performances des Basic me jette la première pierre, s'il ose.

La petite bête qui a eu un succès fou, c'était le ZX 80 puis le ZX 81, parce que c'était pas cher. Et à l'époque, ce qui était pas cher, c'était bien daube-daube. Ceci dit, malgré son unique kilo-octet de RAM,

suite page 2...

GRAND CONCOURS STMAGICIEL & SUZUKI

Voyez en page 35.

Des centaines de petites annonces. Pour vendre ou pour acheter ? Les deux, mon Colonel, et aussi pour échanger, diverser ou employer. Le plus beau, c'est que c'est gratuit, ça coûte pas un radis, c'est ce qu'on appelle une aubaine ! Elles sont quelque part dans le canard.

FORMATION AU 88888
par Fabrice BROCHE. Page 35.

FORMATION AU RISC ET AU CISC
par Sébastien MOUGEY. Page 35.

Ceci est un code-barre. Il est beaucoup plus joli que celui des concurrents, le nôtre est le plus réussi, regardez-moi la grâce de ces lignes, c'est beau. Là, avec un code comme celui-ci, on ne se moque pas du client, on a fait les choses en grand.

M 4058 - 1 H - 20,00 F -RD

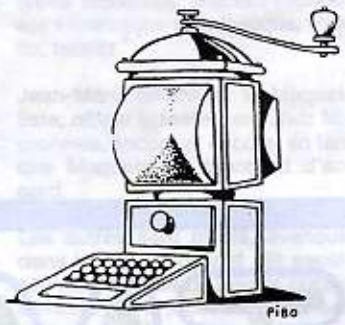
DES PROGRAMMES POUR VOTRE ATARI ST :
GFA BASIC • ASSEMBLEUR • BASIC OMIKRON • LANGAGE C • STOS BASIC

MOD_CONV

GFA-BASIC

Il s'agit d'une moulinette dans laquelle vous passez vos modules soundtracker (.MOD) pour obtenir des fichiers pour Quartet: des échantillons (.AVR) et le voiceset (.SET).

Pascal CRAPONNE & Benoit ARRIBART



Le module choisi est décomposé en échantillons, qui sont passés de la fréquence du module (généralement 8400 Hz) à celle de Quartet (16000 Hz par exemple). Ces deux fréquences sont modifiables (fi% et ff%).

Les échantillons sont transformés un par un en échantillons Quartet, puis le voiceset est créé. Le programme gère les modules de 15 à 31 instruments.

```
REM ******************************************
REM * conversion de modules soundtracker (MOD) *
REM * en échantillons (AVR) et voiceset (SET)  *
REM * pour QUARTET                             *
REM ******************************************
REM *   (c) Les deux sciés de l'écran - 1991   *
REM *    Pascal Craponne et Benoit Arribart    *
REM ******************************************
REM
REM *******************
REM * initialisations *
REM *******************
REM
DIM avr$(30)              !buffers pour samples
DIM avr_name$(30)         !nom des samples
DIM length%(30)           !taille des samples
DIM repeat%(30)           !point de bouclage
DIM repeat_length%(30)    !longueur de bouclage
nb_sample%=-1             !nombre de samples
dummy%=0
fi%=8400                  !fréquence initiale
ff%=16000                 !fréquence finale
REM
REM ***********************
REM * selection du module *
REM ***********************
REM
FILESELECT "*.MOD","",mod$
IF mod$<>""
  @cree_avr               !création des AVR
  @cree_set               !création du SET
ENDIF
END
PROCEDURE cree_avr
  REM
  REM ************************
  REM * ouverture du module *
  REM ************************
  REM
  id$=SPACE$(4)           !type de module
  OPEN "i",#0,mod$
  SEEK #0,&H438
  BGET #0,VARPTR(id$),4
  IF id$="M.K."           !31 instruments
    nb_inst%=31
    SEEK #0,&H3B6
    n%=INP(#0)            !nombre de positions
    SEEK #0,&H3B8
    sam%=&H43C
  ELSE                    !15 instruments
    nb_inst%=15
    SEEK #0,&H1D6
    n%=INP(#0)            !nombre de positions
    SEEK #0,&H1D8
    sam%=&H258
  ENDIF
  FOR i%=0 TO n%          !pour chaque position
    p%=INP(#0)            !on lit quel pattern
    nb_pat%=MAX(p%+1,nb_pat%) !on note le max
  NEXT i%
  ADD sam%,nb_pat%*1024   !offset des samples
  REM
  REM ************************************
  REM * lecture des infos sur les samples *
  REM ************************************
  REM
  PRINT "Exploration du module :"
  SEEK #0,20
  j%=0
  FOR i%=0 TO nb_inst%-1
    avr_name$(j%)=SPACE$(22)
    BGET #0,VARPTR(avr_name$(j%)),22 !nom du sample
    BGET #0,VARPTR(length%(j%))+2,2  !longueur
    BGET #0,VARPTR(repeat%(j%)),4    !volume+repeat
    BGET #0,VARPTR(repeat_length%(j%))+2,2  !repeat
length
    IF length%(j%)>0
      length%(j%)=2*length%(j%)
      repeat%(j%)=2*(repeat%(j%) AND 65535)
      repeat_length%(j%)=2*repeat_length%(j%)
      INC j%
      INC nb_sample%
    ENDIF
  NEXT i%
  PRINT nb_sample%+1;" échantillons."
  REM
  REM ************************
  REM * lecture des samples *
  REM ************************
  REM
  PRINT
  PRINT "Lecture des échantillons :"
  SEEK #0,sam%
  FOR i%=0 TO nb_sample%
    PRINT avr_name$(i%);
    sour$=STRING$(length%(i%),0)
    IF LOC(#0)+length%(i%)>LOF(#0)
      PRINT " : tronqué"
      length%(i%)=LOF(#0)-LOC(#0)
      repeat%(i%)=0
      repeat_length%(i%)=length%(i%)
    ELSE
      PRINT " : ok"
    ENDIF
    BGET #0,VARPTR(sour$),length%(i%)
    REM ** ici on convertit la fréquence **
    REM ** de Fi% Hz à Ff% Hz            **
    length%(i%)=length%(i%)*ff%/fi%
    repeat%(i%)=repeat%(i%)*ff%/fi%
    repeat_length%(i%)=repeat_length%(i%)*ff%/fi%
    avr$(i%)=STRING$(length%(i%),0)
    FOR j%=0 TO length%(i%)-1
      POKE VARPTR(avr$(i%))+j%,PEEK(VARPTR(sour$)
+j%*fi%/ff%)
    NEXT j%
  NEXT i%
  REM
  REM ********************
  REM * ecriture des .AVR *
  REM ********************
  REM
  PRINT
  PRINT "Ecriture des AVR : "
  FOR i%=0 TO nb_sample%
    PRINT avr_name$(i%);" -> ";
    j%=INSTR(avr_name$(i%),":")
    nom_fich$=MID$(avr_name$(i%),j%+1,8)
    DO                          !on remplace les ":"
      j%=INSTR(nom_fich$,":")
      EXIT IF j%=0
      nom_fich$=LEFT$(nom_fich$,j%-1)+"_"+MID$
(nom_fich$,j%+1,LEN(nom_fich$)-j%)
    LOOP
    j%=INSTR(nom_fich$,CHR$(0))
    IF j%=0
      j%=9
    ENDIF
    nom_fich$=UPPER$(LEFT$(nom_fich$,j%-1))
    avr_name$(i%)=nom_fich$
    PRINT avr_name$(i%)+".AVR"
    OPEN "o",#1,avr_name$(i%)+".AVR"
    @cree_header(i%)
    BPUT #1,VARPTR(avr$(i%)),length%(i%)
    CLOSE #1
  NEXT i%
  CLOSE #0
RETURN
PROCEDURE cree_set
  REM
  REM ************************
  REM * header d'un voiceset *
  REM ************************
  REM
  set$=LEFT$(mod$,INSTR(mod$,"."))+"SET"
  PRINT
  PRINT "Création du Voiceset ";set$
  OPEN "O",#1,set$
  OUT #1,16                 ! fréquence
  OUT #1,nb_sample%+2       ! nombre de samples
  FOR i%=0 TO nb_sample%    ! noms des samples
    PRINT #1," "+LEFT$(avr_name$(i%),4)+" "+CHR$(0);
  NEXT i%
  PRINT #1,STRING$(142-LOC(#1),0);
  a%=342
  FOR i%=0 TO nb_sample%    ! adresse des samples
    PRINT #1,MKL$(a%);      ! dans le voiceset
    ADD a%,length%(i%)+28
  NEXT i%
  PRINT #1,STRING$(222-LOC(#1),0);
  REM
  REM *********************
  REM * headers et samples *
  REM *********************
  REM
  FOR i%=0 TO nb_sample%
    @cree_header(i%)
    PRINT "Echantillon : ";avr_name$(i%)
    SEEK #1,LOC(#1)-8
    PRINT #1,MKL$(-1);
    PRINT #1,MKI$(length%(i%));MKI$(0);
    FOR j%=0 TO length%(i%)-1
      POKE VARPTR(avr$(i%))+j%,PEEK(VARPTR(avr$
(i%))+j%)-128
    NEXT j%
    BPUT #1,VARPTR(avr$(i%)),length%(i%)
  NEXT i%
RETURN
PROCEDURE cree_header(i%)
  REM
  REM ************************************
  REM * création de header de fichier AVR *
  REM ************************************
  REM
  PRINT #1,"2BIT";
  PRINT #1,avr_name$(i%)+STRING$(8-LEN(avr_name$
(i%)),0);
  PRINT #1,MKI$(0);         !échantillons mono
  PRINT #1,MKI$(8);         !codés sur 8 bits
  PRINT #1,MKI$(-1);        !signés
  IF repeat_length%(i%)<ff%/1000  !pas de repeat
    repeat%(i%)=0
    repeat_length%(i%)=length%(i%)
    PRINT #1,MKI$(0);       !donc pas boucle
  ELSE
    PRINT #1,MKI$(-1);      !sinon boucle
  ENDIF
  PRINT #1,MKI$(-1);        !corresp. MIDI: inconnue
  PRINT #1,CHR$(3);         !fréq sous Replay: 16168
  PRINT #1,CHR$(0);MKI$(ff%);
  PRINT #1,MKL$(length%(i%));  !long. en octets
  PRINT #1,MKL$(repeat%(i%));  !début de boucle
  PRINT #1,MKL$(repeat_length%(i%)); !long. boucle
  PRINT #1,STRING$(26,0);      !réservés
  REM ** Attention ci-dessous à la longueur de **
  REM ** la chaine qui doit être de 64 octets  **
  PRINT #1,"Les deux sciés de l'écran";SPACE$(39);
RETURN
```

MUSIQUE
Toutes Résolutions

suite de la page 1...

il s'était vendu à plusieurs millions d'exemplaires, le ZX 81. On en file à un jeune, aujourd'hui, il te griffe les joues, ça nous apprendra à faire des expériences débiles, aussi. Bon, il faut reconnaître que fallait quand même être particulièrement secoué pour se payer un truc aussi nullard, même pour moins de 700 francs, ça a marché parce que Sinclair faisait croire qu'on pouvait s'initier à l'informatique en pianotant sur un clavier conçu pour des nabots.

Et puis Sinclair a fait des progrès, ou a eu des scrupules, et a sorti le ZX Spectrum, voilà enfin une machine potable, tant qu'on met de côté le clavier-gomme et le Basic Sinclair. On a commencé à voir des jeux à te mettre l'eau à la bouche, et la machine était bien servie point de vue extensions.

Avant cela, il était sorti le Dragon 32, avec 500 balles (et des brouettes) à raquer en plus pour avoir 32 malheureux Ko supplémentaires. Ca les fait sourire, les jeunes, hein? Et les neuf couleurs, ça les fait éclater de rire, hein?

SI LA MORT EST UN MYSTERE

A cette époque-là, il y avait une tripotée de machines, c'était pas simple de choisir, quoiqu'on en éliminait certaines d'office: l'Aquarius, le Laser 200, Le Jupiter Ace, le BBC, le MC10 de chez Tandy, etc. C'est pas qu'elles fussent mauvaises, mais bon, si on savait qu'elles existaient, elles ne nous attiraient pas, ça se limitait à ça.

En fait, à partir de 84-85, certaines machines sont sorties du peloton, et celui qui avait des brouzoufs devait se creuser la tête pour être sûr de pas se planter, histoire que les voisins ne se moquent pas.

C'est pas difficile: il y avait l'Atari 800XL, le Vic 20 rapidement remplacé par le Commodore 64, l'Apple IIc, les Oric (le simple au goût nature, et l'Atmos), le TI 99/4 A. Après seulement sont arrivés les Amstrad, le QL, et les MSX, puis le ST et l'Amiga ont fait le ménage. Elles ne sont pas toutes sorties en même temps, alors, du coup, on pouvait regretter son choix trois mois après avoir lâché ses économies, les grosses bouboules. Chacune avait ses fanatiques, ce qui compliquait encore plus les choses, y a rien de pire que des avis tranchés et contradictoires quand il s'agit de choisir, dans un domaine où on y connait nib. Ces machines, on les trouvait géniales, et si tu disais le contraire c'était que tu n'y connaissais rien et prends ça, tu l'as bien mérité, et ça, et ça, et ça, salaud.

LA VIE N'A RIEN DE TENDRE

Comme dans tout système, il se produit un phénomène de régulation, d'auto-élimination, les plus faibles disparaissent naturellement. Autrement dit, il y a les flops, capito? Dans ce rayon, l'exemple-type, ça a été le micro-console Adam, de chez CBS Colecovision, le flop de la mort, parce qu'il y avait eu un tel battage médiatique qu'on en a ri à gorge d'autant plus déployée que le succès attendu était nul. A la deuxième place, le Lynx, à ne pas confondre avec la petite console d'Atari (Au passage, messieurs d'Atari France, si vous continuez à dire "Le Lynx" quand tout le monde parle de "La Lynx", ST Magazine ne parlera plus que de "la ST", c'est vu?), un micro-ordinateur que l'on présentait comme un concurrent possible au Commodore 64, vous en avez déjà vu tourner un, vous?

Derrière, le Lansay 64, baptisé Elan dans un premier temps, s'est vautré lamentablement, chose bizarre puisqu'il était pas mal attendu, surtout après les tests dithyrambiques qui l'avaient accueilli, surtout ceux d'Hebdogiciel. Pour finir en beauté, avant de passer au rayon français (quoique flops et produits français ont fait bon ménage), il faut citer le fameux Laser 3000. Ha, vous avez oublié. C'était la bébête qui allait permettre d'avoir tous les bons jeux miam-miam de l'Apple, pour quelques milliers de francs de moins. Manque de bol, ça marchait pas, la machine opposait un niet ferme et définitif devant la majorité des jeux Apple. C'est con, hein?

SI LE CIEL A UN ENFER

C'était l'époque où les français pensaient encore qu'ils pouvaient se faire une place, nous étions les génies de l'informatique. Depuis, vous avez peut-être remarqué, nous ne sommes plus que les génies du logiciel, y a Borland qui monte à la tête. Bref, nous autres français, face au péril étranger, nous avons sorti: l'Hector (avec un Forth en ROM, les étrangers n'y avaient pas pensé, quels crétins ces étrangers tout de même!), l'EXL d'Exelvision (c'était pas de la faute aux français si les étrangers avaient inventé les télévisieurs à télécommande!), l'Alice (flop flop), la gamme TO et MO de chez Thomson (j'arrête, je vais vomir), le Squale (une belle machine blanche et bleue), et, j'ai gardé la meilleure pour la bonne bouche, le Téléstrat. Ca, c'était de la bécane, un vrai bijou, mais que veux-tu, Fabrice, les français, ils aiment avoir des martyrs, et ils t'ont sacrifié sur l'autel de l'incompréhension.

LE CIEL PEUT BIEN M'ATTENDRE

Pour parler de tout ça, on avait plein de journaux, avec un esprit pionnier que ceux d'aujourd'hui sont bien loin de réincarner, aujourd'hui on ne pense qu'au fric, on crée des pin's, on oublie le principal, tout fout le camp. Vous aviez MICRO7, HEBDOGICIEL, LIST, POKEY, L'ORDINATEUR INDIVIDUEL, MICRO SYSTEMES, TILT, SVM, MICRO V.O... Et je vous passe tous les dédiés ("dédiés à une machine ou à un standard"), ça fourmillait.

Par ailleurs, il y avait des émissions à la télé, le grand gourou informatique étant Georges Leclerc, qui tentait de vulgariser un domaine dont personne n'avait rien à battre, ça faisait de ces bides à la télé, une horreur. Ne rigolez pas, il y avait même des émissions sur les radios libres, les animateurs relisaient la presse et diffusaient des programmes à l'antenne, on pouvait les enregistrer sur cassette, ça marchait une fois sur cent.

JE VEUX RETROUVER L'INNOCENCE

Voilà, on a fait à peu près le tour, j'ai volontairement oublié de parler de quelques machines, tant pis. Il est temps de conclure. Tout cet article remarquablement bien construit, avec des exemples particulièrement bien choisis et un fil conducteur passionnant, c'était uniquement pour parler d'anciennes machines, vous voyez? Je veux dire, il ne faut pas chercher de message, il n'y en a pas, c'était juste histoire de causer. A la rigueur, c'était peut-être pour éviter d'avoir à parler de l'avenir de la micro, du ST, et d'autres trucs aussi barbants, ok, j'admets, si ça peut vous faire plaisir. Ciao, et que les Dieux vous protègent.

VEILLEUR 68000

Un programme très court qui va se charger d'éteindre l'écran au bout de deux minutes d'inactivité.

Simon DE GIVRY

```
;*************************************************************
; programme VEILLEUSE (extinction de l'ecran au bout de n minutes)
; auteur  Simon de Givry
;*************************************************************

minutes     equ     2           ;delai de n minutes
            section     text
debut
            move.l      sp,a5
            move.l      4(a5),a5        ;a5=base page
            move.l      $c(a5),d0       ;d0=taille text
            add.l       $14(a5),d0      ;  +taille data
            add.l       $1c(a5),d0      ;  +taille bss
            add.l       #256,d0         ;  +base page
            move.l      d0,taille_prg   ;sauve la taille prg

            clr.l       -(sp)
            move.w      #$20,-(sp)
            trap        #1              ;superviseur
            addq.l      #6,sp

            move.l      d0,save_ssp     ;sauve pile ssp
            move.w      sr,save_sr      ;sauve sr
            move.w      #$2700,sr       ;bloque interruptions

            moveq.l     #8,d1           ;d1=frequence -> 70hz ..
            move.l      #vbl_monochrome,a2
            move.l      #new_clavier_mono,a3
            move.b      $ffff8260.w,d0 ;d0=resolution
            and.b       #%11,d0
            cmp.b       #%10,d0         ;est-on en 70Hz ?
            beq.s       freq
            move.l      #vbl_couleurs,a2
            move.l      #new_clavier_colo,a3
            moveq.l     #4,d1           ;60hz
            move.b      $ffff820a.w,d0  ;d0=mode de synchro
            and.b       #%10,d0
            tst.b       d0              ;est-on en 60Hz ?
            beq.s       freq
            moveq.l     #0,d1           ;50hz
freq
            lea         frequence(pc),a0
            move.l      (a0,d1.w),hertz
;initialise hertz

            clr.l       compteur_vbl    ;initialise le compteur de Vbl
            move.w      #-1,allume      ;écran allumé

            move.l      $118,ancien_clavier
            move.l      a3,$118         ;détourne clavier

            move.l      $456.w,a0       ;a0 pointeur sur adresses VBLs
            lea         4(a0),a0        ;saute le pointeur GEM
            move.w      $454.w,d0       ;nbre max de VBLs
            subq.w      #2,d0           ;pour le dbra .. et le ptr GEM
recherche
            tst.l       (a0)+
            beq.s       trouve
            dbra        d0,recherche
trouve      ;REMARQUE : si pas de place alors ecrase derniere VBL
            move.l      a2,-(a0)        ;installe notre routine vbl
            move.w      save_sr,sr      ;restaure interruptions

            move.l      save_ssp,-(sp)
            move.w      #$20,-(sp)
            trap        #1              ;utilisateur
            addq.l      #6,sp

            move.w      #0,-(sp)
            move.l      taille_prg,-(sp)
            move.w      #$31,-(sp)
            trap        #1              ;keep process
;bye bye ...
;*************************************************************
new_clavier_colo        ;appel midikey puis lance new_clavier
            clr.l       compteur_vbl    ;reinitialise le compteur VBL
            move.l      ancien_clavier(pc),-(sp) ;appel de midikey
            rts
;*************************************************************

new_clavier_mono        ;appel midikey puis lance new_clavier
            clr.l       compteur_vbl    ;reinitialise le compteur VBL
            tst.w       allume          ;ecran eteint ?
            bne.s       continue
            movem.l     d0/d1/a0,-(sp) ;sauve registres dans pile
            lea         $ffff8200.w,a0 ;a0=base reg video
            clr.l       d1
            movep.w     1(a0),d1
            lsl.l       #8,d1           ;d1=debut adr. ecran
attends                 ;attends debut affichage ecran ..
            movep.l     3(a0),d0
            and.l       #$00ffffff,d0   ;d1=compteur video
            cmp.l       d0,d1
            bne.s       attends
            tst.b       $ffff8209.w
            bne.s       attends
            bclr.b      #0,$ffff820a.w ;allume ecran
            move.w      #-1,allume      ;mise a jour de allume
            movem.l     (sp)+,d0/d1/a0 ;restaure registres de la pile
continue
            move.l      ancien_clavier(pc),-(sp) ;appel de midikey
            rts
;*************************************************************
liste_reg_vbl           reg         d0-d6/a0

vbl_couleurs            ;notre routine vbl d'extinction
            move.l      compteur_vbl(pc),d7
allume_ecran
            tst.w       allume          ;ecran eteint ?
            bne.s       eteint_ecran
            tst.l       d7              ;compteur initialise ?
            bne.s       fin_vbl         ;pas d'incrementation
            movem.l     couleurs(pc),liste_reg_vbl
            movem.l     liste_reg_vbl,$ffff8240.w  ;allume l'ecran
            move.w      #-1,allume      ;ecran allumé
            bra.s       incremente
eteint_ecran
            cmp.l       hertz,d7        ;compteur > hertz ?
            blo.s       incremente
            movem.l     $ffff8240.w,liste_reg_vbl
            movem.l     liste_reg_vbl,couleurs   ;sauve les couleurs
            movem.l     eteint(pc),liste_reg_vbl
            movem.l     liste_reg_vbl,$ffff8240.w  ;eteint l'ecran

            clr.w       allume          ;ecran eteint
            bra.s       fin_vbl         ;pas d'incrementation
incremente
            addq.l      #1,compteur_vbl
fin_vbl
            rts
;*************************************************************
vbl_monochrome                          ;notre routine vbl
d'extinction
            move.l      compteur_vbl(pc),d7
            cmp.l       hertz,d7        ;compteur > hertz ?
            blo.s       incremente2
            move.w      sr,save_sr_vbl
            move.w      #$2700,sr       ;bloque les interruptions
            lea         $ffff8200.w,a0 ;a0=base reg video
            clr.l       d1
            movep.w     1(a0),d1
            lsl.l       #8,d1           ;d1=debut adr. ecran
            movep.l     3(a0),d0
            and.l       #$00ffffff,d0   ;d1=compteur video
            cmp.l       d0,d1
            bne.s       restaure_sr
            tst.b       $ffff8209.w     ;au debut de l'ecran ? ...
            bne.s       restaure_sr
            bset.b      #0,$ffff820a.w ;eteint l'ecran
            clr.w       allume          ;ecran eteint
restaure_sr
            move.w      save_sr_vbl,sr ;restaure le sr
            bra.s       fin_vbl2
incremente2
            addq.l      #1,compteur_vbl
fin_vbl2
            rts
;*************************************************************
            section data
frequence
            dc.l        50*60*minutes   ;délai d'effacement 50Hz
            dc.l        60*60*minutes   ;délai d'effacement 60Hz
            dc.l        71*60*minutes   ;délai d'effacement 70Hz

eteint                  ;permet d'annuler les 16 couleurs
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
            dc.w        0
;*************************************************************
            section     bss
compteur_vbl    ds.l    1               ;compteur VBL
ancien_clavier  ds.l    1               ;sauvegarde de la routine
clavier du TOS
hertz           ds.l    1               ;délai avant extinction
allume          ds.w    1               ;booléen sur état de l'écran
                                        ; -1 : allumé , 0 : éteint
save_ssp        ds.l    1               ;sauvegarde pile superviseur
save_sr         ds.w    1               ;sauvegarde du reg. sr
taille_prg      ds.l    1               ;sauvegarde taille du prg
couleurs        ds.w    16              ;sauvegarde des couleurs
save_sr_vbl     ds.w    1               ;sauvegarde du reg. sr ds vbl
```

UTILITAIRE ECRAN
Toutes Résolutions

eddy tôt

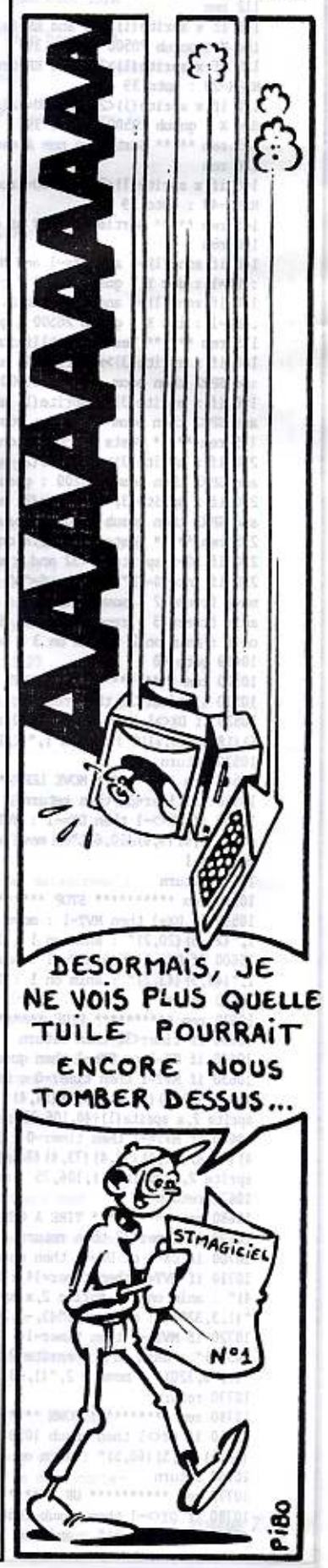
S'il y en a un qui s'est fait avoir en beauté, c'est bibi. Je vous explique, ça vaut le coup : pour moi, la superstition, les réincarnations, c'est salades et compagnie. Les guéridons qui s'envolent, les types qui font parler les morts ou qui tordent les cuillers à distance, ce sont des histoires pour drogués ou pour vieilles folles, ni plus ni moins.

L'exemple le plus fort pour rire de ces débiles profonds, c'est la Malédiction des Pyramides, ça c'était fort. Tu rentres dans une pyramide, paf, tu meurs raide quelques mois après, par la volonté divine d'un Pharaon squeezé 4000 ans auparavant, bravo, j'applaudis, ah, ah, faites m'en d'autres comme celle-là! Eh bé ces cons-là, ils seraient capables de me faire peur, avec leurs histoires de malédiction.

C'est dingue, depuis que nous avons parlé de ce hors-série sauce Hebdogiciel, tout se casse la gueule. D'abord, notre chef PAO se barre en vacances en laissant tous les MAC de la boite avec des claviers danois, sa langue natale. Le danois, c'est bourré de å, de Å, c'est hyper pratique pour les listings, un vrai bonheur. Ensuite, les disques durs du serveur choisissent de fondre au soleil, il faut dépanner. Là-dessus, les maquettistes s'en vont l'un après l'autre, il faut tout faire soi-même pendant qu'ils écoutent du rock, les doigts de pied en éventail. Pour finir, pas moyen de mettre la main sur un directeur de la publication pour censurer ce qui est écrit, il est malade à mourir.

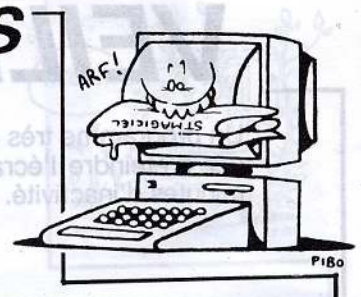
J'ai fini ce canard, je pars en Egypte, laissez-moi vos messages sur 3615 DIVINITEL. Ciao.

Mic Dax



HOT DOG

STOS



C'est malin, vous vous êtes laissé prendre et vous voilà enfermé dans un labyrinthe. Vous ne pouviez pas vous tenir tranquille, non? Bon, voici une mitraillette, débarrassez-vous des monstres enragés et sortez-vous de là, je vous attends dehors.

Antoine LECLERQ et Pierre-Olivier ROTHEVAL

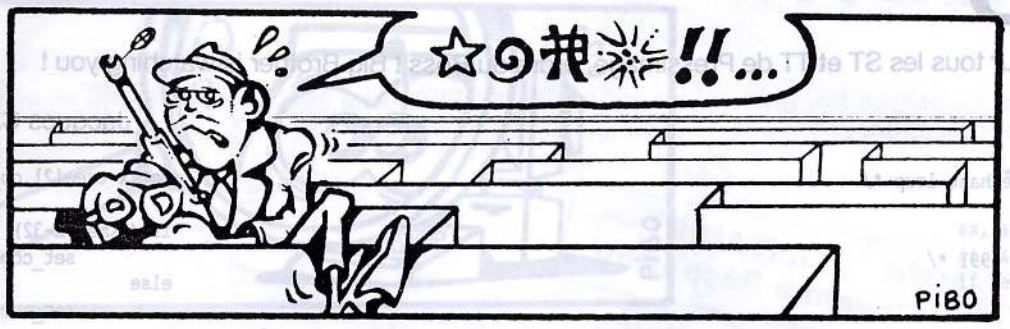
JEU
Basse Résolution

```
1 rem ***********  H O T   D O G  !!! *************
2 rem * (C) 1991 ST MAGAZINE  ET :
3 rem *                  A. LECLERCQ
4 rem *                  P. ROTHEVAL
5 rem ** Initialisations : variables etc... **
6  : dim SAL(5,5,5) : gosub 50000 : load "a:medor.mbs"
7  : mode 0 : key off : curs off : get palette (5)
8 dim ACT(16),VIE(12,2),VIE2(8,2)
9 show : pen 9 : gosub 40300 : gosub 40800 : A$=file select$
("*.DAT","Chargement de carte ?") : hide : if A$="" then gosub
20400 else gosub 41700
10 for C=0 to 15 : colour C,0 : next C : hide on
20 pen 8 : locate 0,0 : centre "Par A. Leclercq & P. Rotheval"
: print : print : centre "(C) ST MAGAZINE & les auteurs 1991" :
X$=screen$(back,0,0 to 319,50)
30 M$="HOT DOG!" : XX=0 : YY=0 : gosub 20200 : screen$(physic,
0,160)=X$ : gosub 40000 : music off
35 LVL=5 : SCR=0 : X=1 : Y=1 : PV=8 : NV=5 : gosub 20500 :
XSPR=100 : NSPR=20 : sprite 2,500,0,1 : music 2
39 sprite 1,XSPR,85,NSPR : wait vbl : set zone 1,140,80 to
160,100
40 rem ***********************************************************
41 rem *********** CORPS DU PROGRAMME *************************
42 rem ***********************************************************
50 if jleft then gosub 10550
60 if jright then gosub 10500
70 if joy=0 then gosub 10580
80 if fire then gosub 10620
90 if zone(1)<>0 and abs(CHLVL)=1 and peek($FFFFC02)=$39 then
LVL=LVL+CHLVL : gosub 20500 : XSPR=160 : NSPR=20 : goto 39
100 if jdown then gosub 10740
110 if jup then gosub 10770
111 rem ***** Sortie ou non à droite ?
112 rem
115 if x sprite(1)>270 and ES=false then XSPR=20 : NSPR=20 :
inc X : gosub 20500 : goto 39
120 if x sprite(1)>270 and ES=true then XSPR=x sprite(1)-3 :
NSPR=20 : goto 39
130 if x sprite(1)<20 and OU=false then XSPR=265 : NSPR=43 :
dec X : gosub 20500 : goto 39
135 rem ***** Sortie ou non à gauche ?
136 rem
140 if x sprite(1)<20 and OU=true then XSPR=x sprite(1)+3 :
NSPR=43 : goto 39
145 rem ***** Sortie en HAUT et en BAS
146 rem
160 if zone(1)=1 and DY=-1 and NO=false then XSPR=150 : NSPR=20
: DX=1 : dec Y : gosub 20500 : goto 39
170 if zone(1)=1 and DY=1 and SU=false then XSPR=150 : NSPR=20
: DX=1 : inc Y : gosub 20500 : goto 39
175 rem ****** Tests de collision entre heros et robot
180 if x sprite(3)>x sprite(1) and x sprite(3)-x sprite(1)<35
and SPR2 then boom : gosub 41000
190 if x sprite(3)<x sprite(1) and x sprite(1)-x sprite(3)<35
and SPR2 then boom : ink 0 : gosub 41000
195 rem ***** Tests de collision entre robot et balle
200 if x sprite(3)>x sprite(2) and x sprite(3)-x sprite(2)<35
and SPR2 then gosub 40200 : gosub 20300
210 if x sprite(3)<x sprite(2) and x sprite(2)-x sprite(3)<35
and SPR2 then gosub 40200 : gosub 20300
215 rem ***** Tests de collision entre Sprite et Goodies
220 if XG-x sprite(1)<32 and x sprite(1)<XG then gosub 40500
230 if inkey$="P" or inkey$="p" then clear key move freeze 1 :
move freeze 2 : move freeze 3 : anim freeze 1 : anim freeze 2 :
anim freeze 3 : repeat : until inkey$="p" or inkey$="P" : anim
on 1 : anim on 2 : anim on 3 : move on 1 : move on 2 : move on 3
10499 goto 40
10500 rem ********** MOVE RIGHT ******************
10510 if timer<32 then return
10520 if DX<>1 then DX=1 : DY=0 : anim 1,"(14,6)(15,6)(16,6)(17,
6)(18,6)(19,6)L" : move x 1,"(1,1,100)L" : anim on 1 : move on 1
10530 return
10540 rem ********** MOVE LEFT ******************
10550 if timer<32 then return
10560 if DX<>-1 then DX=-1 : MVT=1 : DY=0 : anim 1,"(5,6)(6,6)
(7,6)(8,6)(9,6)(10,6)L" : move x 1,"(1,-1,100)L" : anim on 1 :
move on 1
10570 return
10580 rem ********** STOP ******************
10590 if DX=1 then MVT=1 : move off 1 : anim off 1 : anim
1,"(21,5)(20,2)" : anim on 1 : DX=0
10600 if DX=-1 then MVT=-1 : move off 1 : anim off 1 : anim
1,"(44,5)(43,2)" : anim on 1 : DX=0
10610 return
10620 rem ********** FIRE ******************
10630 if timer<32 then return
10640 if DX=1 or DX=-1 then gosub 10580
10650 if MVT=1 then timer=0 : DX=0 : anim 1,"(47,4)(48,4)(49,4)
(50,4)(51,4)(52,4)(53,4)(54,4)" : anim on 1 : move off 2 :
sprite 2,x sprite(1)+40,106,75 : move x 2,"(1, 4,320)" : move on 2
10660 if MVT=-1 then timer=0 : DX=0 : anim 1,"(67,4)(68,4)(69,
4)(70,4)(71,4)(72,4)(73,4)(74,4)" : anim on 1 : move off 2 :
sprite 2,x sprite(1),106,75 : move x 2,"(1,-4,320)" : move on 2
10670 return
10680 rem ********** TIRE A GENOU ******************
10690 if timer<32 then return
10700 if DX=1 or DX=-1 then gosub 10580
10710 if MVT=1 then timer=14 : DX=0 : anim 1,"(22,4)(23,10)(20,
4)" : anim on 1 : sprite 2,x sprite(1)+40,95,75 : move x 2,
"(1,3,320)" : move y 2,"(1,-3,200)" : move on 2
10720 if MVT=-1 then timer=14 : DX=0 : anim 1,"(45,4)(46,10)
(43,4)" : anim on 1 : sprite 2,x sprite(1),95,75 : move x 2,
"(1,-3,320)" : move y 2,"(1,-3,200)" : move on 2
10730 return
10740 rem ********** DOWN ****************
10750 if DY<>1 then gosub 10580 : anim 1,"(55,5)(56,5)( 57,5)
(58,5)(59,5)(60,5)" : anim on 1 : DY=1
10760 return
10770 rem ********** UP ******************
10780 if DY<>-1 then gosub 10580 : anim 1,"(61,5)(61,5)(63,5)
(64,5)(65,5)(66,5)" : anim on 1 : DY=-1
10790 return
11000 rem ** Page de Présentation **
20200 rem ** Procédure d'affichage de page zoomée **
20202 rem ** m$ = texte , xx , yy **
20205 erase 6 : reserve as screen 6 : cls 6 : zoom 5,15,150,49,
189 to 6,60,0,260,199
20210 cls physic : cls back : for C=0 to 15 : colour C,0 : next C
20220 locate 0,0 : pen 4 : paper 0 : print M$
20230 zoom back,0,0,len(M$)*8+1,9 to physic,0,0,319,60 : TXT$=
screen$(physic,0,0 to 320,60)
20240 screen copy 6 to physic : screen$(physic,XX,YY)=TXT$ :
screen copy physic to back :
20250 get palette (6)
20270 return
20300 rem ** Affichage du score **
20310 locate 27,21 : paper 0 : pen 1 : print "      " : locate
27,21 : paper 0 : pen 14 : print "SC:"+str$(SCR)
20320 locate 0,0
20330 wait vbl : return
20400 rem ** Codage des salles **
20410 restore 30010
20415 for LVL=5 to 1 step-1
20420 for Y=1 to 5
20430  : for X=1 to 5
20440 :  : read SAL(LVL,X,Y)
20450  : next X
20460 next Y
20465 next LVL
20470 return
20500 rem ** Décodage des salles **
20501 for I=1 to 15 : doke $FFFF8240+I*2,0 : next I
20502 restore 31000 : for C=1 to 16 : read ACT(C) : next C :
SPR2=false
20503 if X=1 and LVL=1 and Y=1 then music 7 : M$="GAGNE !" :
XX=0 : YY=2 : gosub 20200 : music 7 : clear key : wait key :
cls back : appear back,60 : end
20520 sprite off 3 : sprite off 1 : screen copy 5 to back :
screen copy 5 to physic : gosub 20300
20521 locate 14,23 : print str$(X)-" "+"-"+str$(Y)-" " : home :
NO=false : ES=false : OU=false : SU=false : DY=0 : N=0 : KIND=0 :
CHLVL=0
20522 for C=9 to 11
20523 if btst(C,SAL(LVL,X,Y)) then N=N+2^(C-8)
20524 next C : gosub 40400 : PL=N*10
20525 ink 0 : bar 140,139 to 180,141
20526 ink 1 : box 140,60 to 180,122
20530  :  : for C=0 to 15
20540  :  :  : if btst(C,SAL(LVL,X,Y))=false and ACT(C+1)<>0
then gosub ACT(C+1)
20550  :  : next C
20555 gosub 40900 : gosub 41500 : get palette (5)
20560 return
21000 rem ——— PORTE EN BAS ———
21010 ink 2 : bar 140,139 to 180,141
21020 SU=true : return
21030 rem ——— PORTE EN HAUT ———
21040 ink 1 : bar 140,60 to 180,122
21050 NO=true : return
21060 rem ——— PORTE A DROITE ———
21070 ink 2 : polygon 300,60 to 313,58 to 313,132 to 300,125
21080 ES=true : return
21090 rem ——— PORTE A GAUCHE ———
21100 ink 2 : polygon 18,60 to 6,58 to 6,132 to 18,127
21110 OU=true : return
21200 rem——— PIEUVRE ———
21210 sprite 3,260,110,1
21220 move x 3,"(1,-1,230)(1,1,230)L" : move y 3,"(2,-1,10)(2,
1,10)L" : move on 3
21230 anim 3,"(1,5)(2,5)(3,5)(4,5)L" : anim on 3
21240 SPR2=true : return
21250 rem——— BULLDOG ———
21260 sprite 3,260,100,11 : wait vbl
21270 move off 3 : move x 3,"(1,-1,230)(1,1,230)L" : move on 3
21280 anim 3,"(11,5)(12,5)(13,5)L" : anim on 3
21290 SPR2=true : return
21300 rem——— DROID ———
21310 sprite 3,240,75,87
21320 move x 3,"(1,-1,210)(1,1,210)L" : move on 3
21330 anim 3,"(87,5)(88,5)(89,5)(90,5)(91,5)(92,5)(93,5)(94,5)
(95,5)(96,5)(97,5)(98,5)(99,5)(100,5)(101,5)L" : anim on 3
21340 SPR2=true : return
21400 rem ——— EXTRA LIFE ———
21410 XG=rnd(250)+30
21415 XX=(XG/16)*16-16 : get palette (5) : cls 6 : screen copy
back,XX,115,XX+48,131 to 6,XX,115
21420 ink 5 : KIND=3
21430 earc XG,120,3,4,0,3600
21440 draw XG,123 to XG,128
21450 draw XG-2,125 to XG+2,125
21460 return
21470 rem ——— FORCE ———
21480 XG=rnd(250)+30 : KIND=1
21485 XX=(XG/16)*16-16 : get palette (5) : cls 6 : screen copy
back,XX,115,XX+48,131 to 6,XX,115
21490 ink 14 polygon XG,123 to XG+3,130 to XG-3,130
21500 ink 11 polygon XG,123 to XG+3,130 to XG+1,128
21510 return
21520 rem ——— SOFT FORCE ———
21530 XG=rnd(250)+30 : KIND=2
21535 XX=(XG/16)*16-16 : get palette (5) : cls 6 : screen copy
back,XX,115,XX+48,131 to 6,XX,115
21540 ink 13 : bar XG-3,124 to XG+3,130
21550 ink 15 : polygon XG-3,124 to XG+3,124 to XG+6,122 to XG-
1,122
21560 ink 12 : polygon XG+3,130 to XG+3,124 to XG+6,122 to XG+6
,128
21570 return
30000 rem ** Datas du labyrinthe **
30010 data 29626,29555,21241,29168,29168
30011 data 29172,29168,29180,29168,29168
30012 data 29168,29168,29180,29178,20985
30013 data 29168,29168,15798,29173,25085
30014 data 29168,29168,29168,29154,29173
30015 data 29168,29168,29168,29168,29168
30016 data 29168,29168,29168,29168,29168
30017 data 29168,29168,29168,29168,29160
30018 data 29168,29168,29168,29168,29180
30019 data 29168,29168,29168,29138,22901
30020 data 29168,29168,29168,29168,29168
30021 data 29168,29168,29168,29168,29176
30022 data 29168,29178,29427,22259,29141
30023 data 29168,14780,29168,29168,29168
30024 data 29168,29156,29168,29168,29168
30025 data 29168,29168,29168,29168,29168
30026 data 29168,29168,26554,29153,29168
30027 data 29168,29168,16316,29168,29168
30028 data 29168,29178,29175,29177,29168
30029 data 29170,21719,29171,29173,29168
30030 data 29170,29179,12785,29178,14585
30031 data 29168,32700,29168,29140,30076
30032 data 29170,29183,29171,29171,29173
30033 data 29168,29172,29168,29168,29168
30034 data 29168,29168,29168,29168,29168
30999 rem ** No de lignes selon les portes/méchants/goodies **
31000 data 21090,21060,21030,21000,41100,41200,21300,21200,
21250,0,0,0,21470,21520,21400,0
40000 rem ************* INTRO ************
40010 erase 7 : reserve as screen 7 : cls 7 : logic=7 : def
scroll 1,20,179 to 320,191,-2,0 : Y=190 : LT=1 : S=1 : ST=-1
40030 TEXT$="ET VOICI  H O T  D O G  UN JEU PROGRAMME EN STOS
PAR ANTOINE LECLERCQ (MADMOD) ET P.O ROTHEVAL (PIERROT) EN 24
HEURES CHRONO (ET OUI MIC , ON Y EST ARRIVE!!!)... VEUILLEZ
TOUT D'ABORD LANCER PREPAR.BAS AVANT DE POUVOIR JOUER ... (C)
ST MAGAZINE / LES AUTEURS 1991 ... MAY THE STOS BE WITH YOU !!!"
40035 TEXT$=TEXT$+"  LE PRINCIPE EST SIMPLE : UTILISEZ VOTRE
MITRAILLEUSE POUR RECTIFIER LES DROIDS ET COLLECTEZ LES GOODIES
... ATTENTION , LES PORTES S'OUVRENT ET SE FERMENT AU HASARD ET
VOUS N'AVEZ QUE 3 VIES POUR PARVENIR AU NIVEAU 1 . PRESSEZ
ESPACE POUR MONTER OU DESCENDRE QUAND C'EST POSSIBLE (OVALE
ENTOURANT UNE FLECHE, SUR LE SOL) . LESGOODIES PEUVENT AUGMEN-
TER VOS POINTS DE VIE (CUBES ROSES) , LES METTRE AU MAXIMUM
(CONES JAUNES) ET VOUS DONNER UNE VIE EN PLUS (PAQUERETTE) .
BONNE CHANCE !!!!"
40040 L=len(TEXT$)
40045 music 4
40050 for I=1 to L : A$=mid$(TEXT$,I,1)
40070 if peek($FFFFFC02)=57 then 40110
40080 locate 0,1 : print A$
40090 for G=0 to 8 : screen copy 7,G,7,G+16,16 to logic,319,180
: wait vbl : screen copy logic,16,180,310,190 to physic,20,Y :
scroll 1
40100 next G : next I : goto 40050
40110 logic=default logic : erase 7 : fade 3 : wait vbl : wait
vbl : wait vbl : physic=default physic : back=default back :
return
40200 rem ************ TOUCHE MECHANT *************
40201 if N<=0 then 40220
40202 if N>12 then N=12
40205 ink 0 : paint VIE(N,1),VIE(N,2) : A=SAL(LVL,X,Y)
40210 dec N
40220 if N<=0 then sprite off 3 : SPR2=0 : boom : A=SAL(LVL,X,
Y) : bclr 9,A : bclr 10,A : bclr 11,A : bset 6,A : bset 7,A :
bset 8,A : SAL(LVL,X,Y)=A
40230 move off 2 : sprite 2,500,0 : wait vbl : boom : SCR=SCR+PL
40240 return
40300 rem ********** INIT TABLEAU DE VIE MECHANT
***************
40310 restore 40350
40320 for I=1 to 12
40330  : read VIE(I,1),VIE(I,2)
40340 next I : return
40350 data 183,183,198,183,183,177,198,177,183,172,198,172,
183,166,198,166,183,160,198,160,183,154,198,154
40400 rem ******** INSTAL VIE MECHANT ********************
40405 if N=0 or N>12 then 40440
40410 ink 12 : for I=1 to N
40420  : paint VIE(I,1),VIE(I,2)
40430 next I
40440 return
40500 rem ********* TAKE GOODY *************************
40505 XX=(XG/16)*16-16
40510 screen copy 6,XX,115,XX+48,131 to back,XX,115
40520 if KIND=1 then PV=8
40530 if KIND=2 and PV<8 then inc PV : gosub 40900
40540 if KIND=3 and NV<5 then inc NV : ink 4 : bar 293,5 to
293+NV*4,7
40550 A=SAL(LVL,X,Y) : bset (11+KIND),A : SAL(LVL,X,Y)=A :
return
40600 rem ******** TU PERDS DES POINTS DE VIE MON POTE ! ***
40610 if PV=0 then dec NV : PV=8 : gosub 40900 : gosub 41500
40620 if NV<=0 then goto 40700
40630 return
40700 rem ********** GAME OVER *********
40710 XX=0 : YY=0 : M$="GAME OVER!" : gosub 20200 : pop : wait
key : goto 9
40800 rem ********** INIT TABLEAU DE VIDE GENTIL ********
40805 restore 40850
40810 for I=1 to 8
40820  : read VIE2(I,1),VIE2(I,2)
40830 next I
40840 return
40850 data 71,179,89,179,71,171,89,171,71,165,89,165,71,159,89,159
40900 rem ******* INSTALLE TABLEAU DE VIE GENTIL ******
40910 for I=1 to PV
40920  : ink 8 : paint VIE2(I,1),VIE2(I,2)
40930 next I : if PV=8 then 40940
40931 for I=PV+1 to 8
40932  : ink 0 : paint VIE2(I,1),VIE2(I,2)
40933 next I
```

suite en page 5

HOT DOG

Il faut encore taper tout ça, et ce n'est pas fini : vous allez devoir taper le second listing, qui sert à préparer les banques comme il faut pour que tout marche bien-bien. Courage.



suite de la page 4

```
40940 return
41000 rem ************ TOUCHE GENTIL ************
41010 ink 0 : auto back off : paint VIE2(PV,1),VIE2(PV,2) :
auto back on
41020 dec PV : gosub 40900
41022 if PV<=0 then dec NV : PV=8 : gosub 40900 : gosub 41500
41025 rem if MVT=1 then sprite 1,x sprite(1)-5,85
41026 rem if MVT=-1 then sprite 1,x sprite(1)+5,85
41030 return
41100 rem *********** Vers le haut ***********
41110 ink 6 : earc 160,130,15,5,0,3600 : draw 160,133 to
160,130 : polyline 157,130 to 163,130 to 160,127 to 157,130
41120 CHLVL=-1 : return
41200 rem *********** Vers le bas  ***********
41210 ink 6 : earc 160,130,15,5,0,3600 : draw 160,127 to
160,130 : polyline 157,130 to 163,130 to 160,133 to 157,130
41222 CHLVL=1 : return
41300 rem ********** OUVERTURE / FERMETURE DES PORTES ********
41310 rem ********** instructions **********
41320 XX=0 : YY=160 : M$="INSTRUCTIONS" : gosub 20200
41330 wait key
41340 cls : mode 1 : key off : curs off : hide
41350 windopen 1,1,1,79,11,0,3 : pen 3
41360 centre "HOT DOG" : print : print : print
41370 centre "Vous devez sortir du labyrinthe de l'infame Blounga"
41380 print : centre "Pour cela , tuez robots , pieuvres et
droids" : print
41390 centre "Et ramassez les Goodies (extra life,points de vie...)"
41400 print : centre "Parvenez au niveau 1 ou se trouve la
sortie..."
41410 print : centre "Utilisez le joystick pour vous déplacer"
: print
41420 centre "Et espace pour monter ou descendre d'un niveau"
41430 wait key
41440 mode 0 : key off : curs off : hide : get palette (5)
41450 pen 5 : locate 0N12 : print "EXTRA LIFE : " : XG=160 :
gosub 21415 : wait key : return
41500 rem ************* AFFICHE NOMBRE DE VIES *******
41510 ink 0 : bar 293,5 to 314,7 : ink 9 : bar 293,5 to
293+NV*4,7
41520 return
41600 rem ********** CHOIX : JEU/EDITEUR/CHARGER UNE MAP ******
41700 rem ********** CHARGER UNE MAP *******
41705 open in #1,A$
41710 for LVL=5 to 1 step-1
41720  : for Y=1 to 5
41730  :  : for X=1 to 5
41740  :  :  : input #1,SAL(LVL,X,Y)
41750  :  : next X
41760  : next Y
41770 next LVL
41780 close #1
41790 return
50000 rem ******************************************************
50010 rem ******* CREATEUR DES MAPS ***************************
50020 rem ******************************************************
50040 mode 0 : key off : curs off : show
50050 windopen 1,0,0,40,12,0,3 : curs off
50060 pen 14
50070 home : print : print
50080 centre "VOULEZ VOUS" : print : print : print : centre
"   CREER UN NOUVEAU JEU —— OUI" : print
50090 centre "CHARGER UN JEU PREEXISTANT - OUI"
50100 set zone 1,33*8,5*16 to 36*8,5*16+16
50110 set zone 2,33*8,6*16 to 36*8,6*16+16
50115 repeat
50120 if zone(0)=1 and FGH<>1 then locate 33,5 : pen 15 : print
"OUI" : locate 33,6 : pen 14 : print "OUI" : FGH=1
50125 if zone(0)=2 and FGH<>2 then locate 33,6 : pen 15 : print
"OUI" : locate 33,5 : pen 14 : print "OUI" : FGH=2
50130 until mouse key<>0
50140 if FGH=2 then return
51000 mode 0 : key off : curs off : LVL=5
51010 home : print "LEVEL  :"
51020 locate 8,0 : print LVL
51030 ink 10 : polygon 20,180 to 30,175 to 30,185
51040 ink 11 : polygon 40,180 to 30,175 to 30,185
51050 dim CO(25,2),A$(5,50,20) : LVL=5
51060 for LVL=1 to 5 : for I=1 to 25 : for U=1 to 15 :
A$(LVL,I,U)="non" : next U : next I : next LVL : LVL=5
51070 I=0 : ink 1
51080 for X=1 to 5
51090 for Y=1 to 5
51100 box X*20,Y*20 to X*20+20,Y*20+20
51110 inc I : set zone I,X*20,Y*20 to X*20+20,Y*20+20
51120 CO(I,1)=X : CO(I,2)=Y
51130 next Y : next X
51140 locate 16,0 : print "salle —"
51150 pen 15 : locate 16,2 : print "porte gauche —"
51160 locate 16,3 : print "porte droite —"
51170 locate 16,4 : print "porte haut ——"
51180 locate 16,5 : print "porte bas ——"
51190 pen 14 : locate 16,6 : print "mvt vers le haut-"
51200 locate 16,7 : print "mvt vers le bas -"
51210 pen 13 : locate 16,8 : print "robot ———"
51220 locate 16,9 : print "pieuvre ———"
51230 locate 16,10 : print "bulldog ———"
51240 locate 16,11 : print "nombre ———"
51250 locate 16,12 : print "  de  fois ——"
51260 locate 16,13 : print " a toucher —— "
51270 pen 12 : locate 16,14 : print " pt de vie max - "
51280 locate 16,15 : print " point de vie — "
51290 locate 16,16 : print " extra life —— "
51300 locate 16,17 : print "to save : click->sav"
51305 locate 16,18 : print "to QUIT : click->QUI"
51310 if mouse key=2 then goto 51310
51330 locate 1,19 : print "choisir une salle"
51340 I=0 : for X=1 to 5
51350 for Y=1 to 5
51360 inc I : set zone I,X*20,Y*20 to X*20+20,
Y*20+20
51370 next Y : next X
51380 if mouse key=0 then 51380
51390 locate 1,19 : print "                    "
51400 T=zone(0)
51410 if T=0 then goto 51330
51420 locate 20,0 : print CO(T,1),CO(T,2)
51430 pen 1 : for I=1 to 15 : locate 33,I+1 : print
A$(LVL,T,I) : next I
51440 for I=1 to 17 : set zone I,33*8,I*8+8 to
33*8+8*4,I*8+16 : next I
51450 if mouse key=2 then pen 1 : goto 51310
51460 if mouse key<>1 then goto 51450
51470 C=point(x mouse,y mouse) : if C=10 or C=11
then goto 51760
51480 bell : U=zone(0) : if U=0 or U>17 then goto
51440
51490 if U=16 then gosub 52010 : goto 51440
51495 if U=17 then return
51500 pen 2 : locate 33,U+1 : print A$(LVL,T,U)
51510 F$=inkey$
51520 pen 1 : if F$="o" or F$="O" then goto 51570
51530 if U>4 then if F$="n" or F$="N" then locate
33,U+1 : print "non" : A$(LVL,T,U)="non" : OK=1
51540 if U<=4 then if F$="n" or F$="N" then goto
51950
51550 if OK<>1 then goto 51510
51560 OK=0 : ink 1 : goto 51450
51570 rem ******* test de validité ****************
51580 if U=5 then shoot : if A$(LVL,T,6)="non" then
locate 33,6 : print "oui" : A$(LVL,T,U)="oui" else
gosub 51720
51590 if U=6 then shoot : if A$(LVL,T,5)="non" then
locate 33,7 : print "oui" : A$(LVL,T,U)="oui" else
gosub 51720
51600 if U=7 then shoot : if A$(LVL,T,8)="non" and
A$(LVL,T,9)="non" then locate 33,8 : print "oui" :
A$(LVL,T,U)="oui" else gosub 51720
51610 if U=8 then shoot if A$(LVL,T,7)="non" and
A$(LVL,T,9)="non" then locate 33,9 : print "oui" :
A$(LVL,T,U)="oui" else gosub 51720
51620 if U=9 then shoot if A$(LVL,T,7)="non" and
A$(LVL,T,8)="non" then locate 33,10 : print "oui" :
A$(LVL,T,U)="oui" else gosub 51720
51630 if U=3 and T<>1 and T<>6 and T<>11 and T<>16
and T<>21 then pen 1 : locate 33,U+1 : print "oui"
: A$(LVL,T,U)="oui" : gosub 51820
51640 if U=4 and T<>5 and T<>10 and T<>15 and T<>20
then pen 1 : locate 33,U+1 : print "oui" :
A$(LVL,T,U)="oui" : gosub 51820
51650 if U=2 and T<21 then pen 1 : locate 33,U+1 :
print "oui" : A$(LVL,T,U)="oui" : gosub 51820
51660 if U=1 and T>5 then pen 1 : locate 33,U+1 :
print "oui" : A$(LVL,T,U)="oui" : gosub 51820
51670 if U=13 then shoot : if A$(LVL,T,14)="non"
and A$(LVL,T,15)="non" then locate 33,14 : print
"oui" : A$(LVL,T,U)="oui" else gosub 51720
51680 if U=14 then shoot if A$(LVL,T,13)="non" and
A$(LVL,T,15)="non" then locate 33,15 : print "oui"
: A$(LVL,T,U)="oui" else gosub 51720
51690 if U=15 then shoot if A$(LVL,T,13)="non" and
A$(LVL,T,14)="non" then locate 33,16 : print "oui"
: A$(LVL,T,U)="oui" else gosub 51720
51700 if U>9 and U<13 then pen 1 : locate 33,U+1 :
print "oui" : A$(LVL,T,U)="oui"
51710 OK=1 : goto 51550
51720 rem *********** erreur *********************
51730 locate 33,U+1 : pen 1 : print "non" :
A$(LVL,T,U)="non"
51740 locate 1,19 : pen 4 : print "impossible
d'avoir deux objets de meme type dans la meme salle
: press key " : bell : wait key : locate 1,19 :
print space$(80)
51750 return
51760 rem ******* change level ********************
51770 if C=10 and LVL>1 then dec LVL
51780 if C=11 and LVL<5 then inc LVL
51790 if mouse key=1 then 51790
51800 locate 8,0 : print LVL
51810 goto 51330
51820 rem ********** draw les portes existantes
************
51830 W=0 : for X=1 to 5
51840 for Y=1 to 5
51850 ink 1 : box X*20,Y*20 to X*20+20,Y*20+20
51860 next Y : next X
51870 W=0 : for X=1 to 5 : for Y=1 to 5
51880 inc W
51890 if A$(LVL,W,3)="oui" then A$(LVL,W-1,4)="oui"
: ink 0 : draw X*20+5,Y*20 to X*20+15,Y*20
51900 if A$(LVL,W,1)="oui" then A$(LVL,W-5,2)="oui"
: ink 0 : draw X*20,Y*20+5 to X*20,Y*20+15
51910 if A$(LVL,W,2)="oui" then A$(LVL,W+5,1)="oui"
: ink 0 : draw X*20+20,Y*20+5 to X*20+20,Y*20+15
51920 if A$(LVL,W,4)="oui" then A$(LVL,W+1,3)="oui"
: ink 0 : draw X*20+5,Y*20+20 to X*20+15,Y*20+20
51930 next Y : next X
51940 return
51950 rem ************* close door
*************************
51960 if U=3 and T<>0 and T<>1 and T<>6 and T<>11
and T<>16 and T<>21 then pen 1 : locate 33,U+1 :
print "non" : A$(LVL,T,U)="non" : A$(LVL,T-1,4)
="non" : gosub 51820
51970 if U=4 and T<>5 and T<>10 and T<>15 and T<>20
then pen 1 : locate 33,U+1 : print "non" : A$(LVL,
T,U)="non" : A$(LVL,T+1,3)="non" : gosub 51820
51980 if U=2 and T<21 then pen 1 : locate 33,U+1 :
print "non" : A$(LVL,T,U)="non" :
A$(LVL,T+5,1)="non: gosub 5000"
51990 if U=1 and T>5 then pen 1 : locate 33,U+1 :
print "non" : A$(LVL,T,U)="oui" : A$(LVL,T-5,2)=
"non" : gosub 51820
52000 U=0 : T=0 : goto 51450
52010 rem *********** transforme en binaire *******
52020 OLDLVL=LVL : for LVL=5 to 1 step-1
52030  : for S=1 to 25
52040  : A=0
52050  :  : if A$(LVL,S,1)="non" then bclr 0,A else
bset 0,A
52060  :  : if A$(LVL,S,2)="non" then bclr 1,A else
bset 1,A
52070  :  : if A$(LVL,S,3)="non" then bclr 2,A else
bset 2,A
52080  :  : if A$(LVL,S,4)="non" then bclr 3,A else
bset 3,A
52090  :  : if A$(LVL,S,5)="non" then bset 4,A else
bclr 4,A
52100  :  : if A$(LVL,S,6)="non" then bset 5,A else
bclr 5,A
52110  :  : if A$(LVL,S,7)="non" then bset 6,A else
bclr 6,A
52120  :  : if A$(LVL,S,8)="non" then bset 7,A else
bclr 7,A
52130  :  : if A$(LVL,S,9)="non" then bset 8,A else
bclr 8,A
52140  :  : if A$(LVL,S,10)="non" then bclr 9,A
else bset 9,A
52150  :  : if A$(LVL,S,11)="non" then bclr 10,A
else bset 10,A
52160  :  : if A$(LVL,S,12)="non" then bclr 11,A
else bset 11,A
52170  :  : if A$(LVL,S,13)="non" then bset 12,A
else bclr 12,A
52180  :  : if A$(LVL,S,14)="non" then bset 13,A
else bclr 13,A
52190  :  : if A$(LVL,S,15)="non" then bset 14,A
else bclr 14,A
52200  : SAL(LVL,CO(S,1),CO(S,2))=A
52210  : next S
52220 next LVL : A$=file select$("*.dat","Sauver la
carte")
52230 open out #1,A$
52240 for LVL=1 to 5
52250  : for Y=1 to 5
52260  :  : for X=1 to 5
52270  :  :  : print #1,SAL(LVL,X,Y)
52280  :  : next X
52290  : next Y
52300 next LVL
52310 close #1
52320 LVL=OLDLVL : return
```

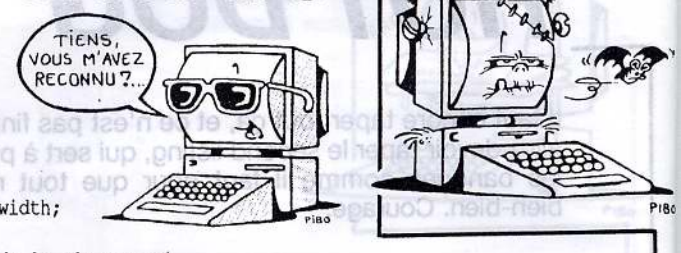
```
10 goto 20200
20000 rem ** Prépare l'écran compacté
20005 rem ** pour nos besoins
20010 mode 0 : key off : curs off : hide
20015 for C=0 to 15 : colour C,0 : next C
20017 locate 0,0 : paper 2 : pen 14 : centre "H O T
D O G !" : A$=screen$(back,40,0 to 250,8)
20020 unpack 11,physic : get palette (physic)
20030 screen copy physic to back : erase 5 : reser-
ve as datascreen 5 : screen copy physic to 5
20050 ink 0 : bar 40,28 to 170,70
20070 ink 1 : polyline 153,21 to 153,27
20080 ink 2 : bar 60,1 to 230,11
20090 ink 1 : polyline 60,0 to 230,0
20100 ink 4 : polyline 60,12 to 230,12
20110 screen$(physic,40,3)=A$
20120 screen copy physic to back
20130 ink 0 : bar 215,159 to 305,180
20140 ink 12 : box 215,159 to 305,180 : ink 0 : bar
110,180 to 142,191
20150 ink 13 : box 216,160 to 304,179 : ink 8 : bar
65,156 to 79,161
20170 erase 11 : erase 5 : reserve as datascreen 5
: screen copy back to 5 : return
20200 rem ** INSTRUCTIONS **
20210 mode 1
20220 key off : hide
20230 windopen 1,0,0,79,12,0,3
20240 centre "Programme de préparation"
20250 print : print : centre "En premier lieu ,
sous l'editeur de sprites" : print
20260 centre "Veuillez :  - Charger Animals1.mbk"
20270 print : centre "          - Merger
Animals1.mbk"
20280 print : centre "          - Merger
Animals2.mbk"
20290 print : centre "          - Merger
Animals2.mbk"
20300 print : centre "(Tous ces fichiers sont
présents sur le disk 'Accessoires')"
20310 print : centre "Puis sauvez les sprites sous
le nom : SPRITE.MBK"
20330 wait key
20340 windel 1
20350 locate 0,12 : centre "Insérez le disk acces-
soires SVP"
20360 print : centre "(Je cherche BACKGRND.MBK &
TUNES.MBK)"
20365 wait key
20370 erase 11 : load "A:BACKGRND.MBK",11 : load
"A:TUNES.MBK"
20380 gosub 20000 : cls : mode 1 : key off
20390 locate 0,12 : centre "Insérez le disk conte-
nant votre fichier"
20400 print : centre "SPRITE.MBK" : print : centre
"(Créé en suivant les instructrions données)"
```

→ page 7

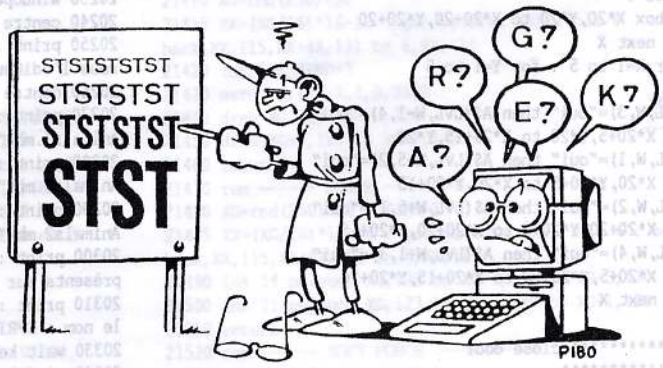
EYES

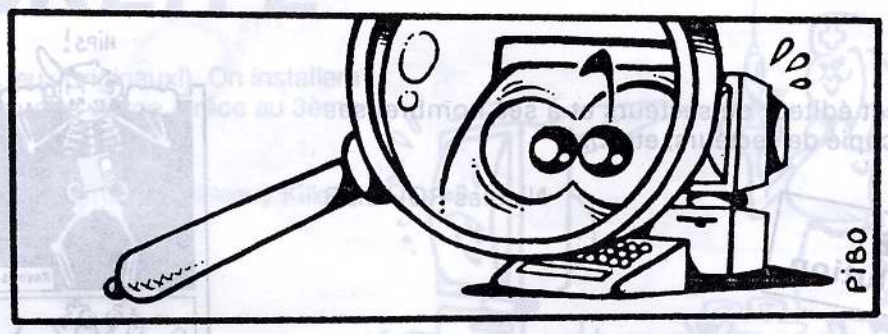
Cet accessoire tourne sur tous les ST et TT de Pressimage, ordre du Boss ! Big Brother is watching you !

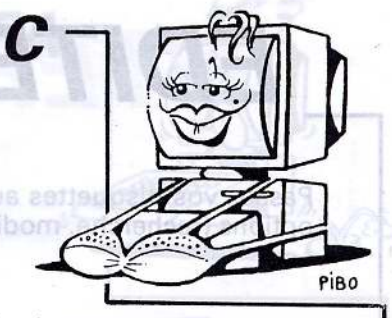
Jacques CARON




```
/* Les ptits yeux du grand méchant loup */

/* Turbo C 2.03 - 11/6/91 */
/* (c) J. Caron & Pressimage 1991 */

#include <stdio.h>
#include <math.h>
#include <aes.h>
#include <vdi.h>
#include <tos.h>

#define RAYON 3

int big_one[16]=  /* le dessin des grands cercles */
{
        0x0000,  0x07c0,  0x1830,  0x2008,  0x4004,  0x4004,  0x8002,  0x8002,
        0x8002,  0x8002,  0x8002,  0x4004,  0x4004,  0x2008,  0x1830,  0x07c0
};

int little_one[5]=        /* les petits (pupilles) */
{       0x7000, 0xf800, 0xf800, 0xf800, 0x7000 };

/* les MFDBs pour les copies de blocs */

MFDB big={ big_one, 16, 16, 1, 1, 1 };
MFDB little={ little_one, 16, 5, 1, 1, 1 };
MFDB scr={ 0 };
/* fd_addr=0 : le VDI trouve tout seul les caractéristiques de l'écran */

/* les couleurs: 1 (noir) et 0 (blanc) */
int colors[2]={ 1, 0 };

/* pour le détournement du trap */
extern void (*o_trap2)();
extern void n_trap2();

int vdih,max_x,ok=0;
int xpos,ypos;            /* coin haut gauche des yeux */
int xc[2],yc;             /* centre des yeux */
int px,py,ox[2],oy[2];    /* ancienne position souris / yeux */
void *screen;             /* l'adresse de l'écran */
int fx,fy,fw,fh;  /* les coords de la boite d'infos */

/* Description de ma "boite de dialogue" ! */

char *string1="Eyes 1.1 - (c) J. Caron 1991";

OBJECT form[]=
{
        -1, -1, -1, G_BUTTON, 0, OUTLINED, 0, 0, 0, 30, 3,
};

/* on initialise le VDI */

void init_vdi(void)
{
        int work_in[11],work_out[57];
        int i;

        vdih=graf_handle(&i,&i,&i,&i);      /* le handle de l'aes, merci */
        work_in[0]=Getrez()+2;              /* périphérique à ouvrir (ou 1) */
        for (i=0;i<10;work_in[i++]=1);
        work_in[10]=2;
        v_opnvwk(work_in,&vdih,work_out);   /* on ouvre */
        max_x=work_out[0];                  /* on note la largeur */
}

/* en sortant (si pas en acc) */

void exit_vdi(void)
{
        v_clsvwk(vdih);
}
/* prépare la boite de dialogue */

void init_box(void)
{
        rsrc_obfix(form,0);                 /* transformations coords */
        form[0].ob_spec.free_string=string1;/* le texte */
        form_center(form,&fx,&fy,&fw,&fh);  /* on centre */
}

/* pour indiquer les coordonnées des yeux */

void set_coords(int x,int y)
{
        ok=1;

        xpos=x;
        ypos=y;
        xc[0]=x+7;
        xc[1]=x+23;
        yc=y+8;
        screen=Logbase();
}

/* prend un arbre d'objets de type menu, et cherche la place où mettre
   les yeux, juste après le dernier menu */

void scan_tree(OBJECT *tree)
{
        /* tree[2]:
                objet contenant les titres de menus,
                fils de la barre de menus (item #1),
                commençant en x=0 dans l'écran (item #0) */
        int x=tree[2].ob_x+tree[2].ob_width;

        if (x<max_x-32)             /* de la place ? */
                set_coords(x,0);
        else
                set_coords(tree[2].ob_x,0);
}

/* Quand on n'a pas de menu, on appelle no_menu, qui met les yeux à la
   droite de l'écran */

void no_menu(void)
{
        set_coords(max_x-32,0);
}

/* la fonction la plus utile : affichage des yeux */

void update_eyes(int x,int y)
{
        int pxy[8];
        int i,dist;
        long xx,yy;
        int nx,ny;
        int hidden=0;

        /* si pas d'yeux en ce moment, ou si on n'a pas bougé,
           ou si on n'est pas sur le bon écran, tant pis */
        if (!ok || (ok==2 && px==x && py==y) || screen!=Logbase())
                return;
        /* NDC: Comment ça, tant pis? */
        /* L'auteur: oui, tant pis, na, c'est comme ça. */
        /* NDC: Ha d'accord, on voit l'genre du mec... */

        /* si la souris est un peu trop près, on la cache */
        if (y<32 || py<32)
        {
                v_hide_c(vdih);
                hidden=1;
        }
        /* pour chacun des yeux */
        for (i=0;i<2;i++)
        {
                /* position relativement au centre de l'oeil */
                xx=x-xc[i];
                yy=y-yc;

                /* distance */
                dist=(int)sqrt(xx*xx+yy*yy);

                /* si à l'extérieur, on se met sur le bord */
                if (dist>=RAYON)
                {
                        nx=xc[i]+RAYON*xx/dist;
                        ny=yc+RAYON*yy/dist;
                }
                else    /* sinon, à la position de la souris */
                {
                        nx=x;
                        ny=y;
                }
                /* centre -> coin supérieur gauche */
                nx-=2;
                ny-=2;

                /* pas bougé ? on s'en va */
                if (ok==2 && nx==ox[i] && ny==oy[i])
                        continue;
                /* NDC: Comment ça, on s'en va? */
                /* L'auteur: taaaaa mèèère! */

                /* le grand cercle */
                pxy[0]=0;
                pxy[1]=0;
                pxy[2]=15;
                pxy[3]=15;
                pxy[4]=xpos+i*16;
                pxy[5]=ypos;
                pxy[6]=xpos+i*16+15;
                pxy[7]=ypos+15;
                vrt_cpyfm(vdih,1,pxy,&big,&scr,colors);

                /* le petit */
                pxy[0]=0;
                pxy[1]=0;
                pxy[2]=4;
                pxy[3]=4;
                pxy[4]=nx;
                pxy[5]=ny;
                pxy[6]=nx+4;
                pxy[7]=ny+4;
                vrt_cpyfm(vdih,1,pxy,&little,&scr,colors);
                ox[i]=nx;
                oy[i]=ny;
        }
        if (hidden)
                v_show_c(vdih,1);
        px=x;
        py=y;
        ok=2;   /* 2=on est déjà passé par là */
}
```








```
/* met en place notre trap 2 à nous */

void init_trap(void)
{
        o_trap2=Setexc(0x22,n_trap2);
}
int main(void)
{
        if (_app)       /* application ?*/
        {
                /* on met les yeux en haut à gauche, et on attend un clic */
                appl_init();
                graf_mouse(ARROW,0L);
                init_vdi();
                set_coords(0,0);
                while (1)
                {
                        int e,x,y,i;

                        e=evnt_multi(MU_TIMER|MU_KEYBD|MU_BUTTON,
                                1,      1,      1,
                                0,      0,      0,      0,      0,
                                0,      0,      0,      0,      0,
                                0,
                                10,     0,
                                &x,     &y,
                                &i, &i, &i,     &i);
                        if (e&(MU_BUTTON|MU_KEYBD))
                                break;
                        update_eyes(x,y);
                }
                exit_vdi();
                appl_exit();
        }
        else    /* accessoire ! */
        {
                int ap_id;

                /* on initialise plein de beau monde */
                ap_id=appl_init();
                menu_register(ap_id," Eyes 1.1");       /* dans menu */
                init_vdi();
                init_box();
                init_trap();

                /* ze boucle */
                while (1)
                {
                        int e,x,y,i,buf[8];

                        /* on attend un événement */
                        e=evnt_multi(MU_TIMER|MU_MESAG,
                                0,      0,      0,
                                0,      0,      0,      0,      0,
                                0,      0,      0,      0,      0,
                                buf,
                                10,     0,
                                &x,     &y,
                                &i, &i, &i,     &i);
                        if (e&MU_MESAG)
                        {
                                switch(buf[0])
                                {
                                        case AC_OPEN:   /* clic sur "Eyes 1.1" */
                                                /* on prend la main */
                                                wind_update(BEG_UPDATE);
                                                /* on réserve */
                                                form_dial(0,0,0,0,0,fx,fy,fw,fh);
                                                /* on affiche */
                                                objc_draw(form,0,10,fx,fy,fw,fh);
                                                /* on attend un clic, tout en bougeant
                                                   les yeux */
                                                while (1)
                                                {
                                                        e=evnt_multi(
                                                        MU_TIMER|MU_KEYBD|MU_BUTTON,
                                                        1, 1, 1,
                                                        0, 0, 0, 0, 0,
                                                        0, 0, 0, 0, 0,
                                                        0,
                                                        10, 0,
                                                        &x, &y,
                                                        &i, &i, &i, &i);
                                                        if (e&(MU_BUTTON|MU_KEYBD))
                                                                break;
                                                        update_eyes(x,y);
                                                }

                                                /* redraw, sioupli */
                                                form_dial(3,0,0,0,0,fx,fy,fw,fh);
                                                /* on rend la main */
                                                wind_update(END_UPDATE);
                                                break;

                                        case AC_CLOSE: /* début ou fin d'application */
                                                ok=0; /* On n'affiche plus les yeux */
                                                break;
                                }
                        }
                        /* si pas ok, on attend un menu_bar jusqu'à 1 seconde */
                        for (i=0;!ok && i<10;i++)
                                evnt_timer(100,0);
                        /* toujours pas ok? On met les yeux à droite */
                        if (!ok)
                                no_menu();

                        /* et le plus important ! */
                        update_eyes(x,y);
                }
                return 0;
}
```

UN SUPER GAG! Toutes Résolutions

RENCONTRER OYA SHIVO ? 3615 STMAG

```
LE LISTINGUE ASSEMBLEUR "TRAP2.S"

; Ici on prend le trap #2, et on récupère menu_bar

        .import  scan_tree,no_menu,ok
        .export  n_trap2,o_trap2

n_trap2:
        cmpi.w   #$c8,d0            ; AES ?
        bne      cont_trap          ; non, cassos (NDC: Hein?)

        movea.l  d1,a0              ; aespb
        movea.l  (a0),a1            ; contrl
        cmpi.w   #30,(a1)           ; menu_bar ?
        bne      cont_trap          ; non, cassos

        movea.l  8(a0),a1           ; intin
        tst.w    (a1)               ; ON ou OFF ?
        beq      cont_trap          ; OFF -> cassos

menu_on:                            ; ON
        movem.l  d0-d7/a0-a6,-(sp)  ; on sauve un peu...

        movea.l  16(a0),a1          ; addrin
        move.l   (a1),a0            ; adresse de l'arbre
        ; NB: pour un autre compilo que TC, il faudra mettre
        ; (a1) sur la pile plutot que dans a0

        jsr      scan_tree          ; fonction C

        ; si on met (a1) sur la pile, ne pas oublier
        ; un addq.l #4,sp ici...

        movem.l  (sp)+,d0-d7/a0-a6  ; on remet tout

cont_trap:
        dc.w     $4ef9              ; JMP
o_trap2:
        dc.l     0                  ; l'ancien TRAP
        illegal
```

```
FICHIER "EYES.PRJ"

c:\eyes.acc

=

TCSTART.O

EYES.C
TRAP2.S

TCSTDLIB.LIB
TCGEMLIB.LIB
TCTOSLIB.LIB
TCFLTLIB.LIB
```

HOTDOG

suite de la page 5

```
20410 wait key : load "a:sprite.mbk" : cls : centre
"MERCI!"
20510 mode 0 : key off : curs off : hide
20520 for I=43 to 46 : sprite 1,100,100,I : screen
copy back to physic : screen copy physic to back :
for X=0 to 47 : gosub 20540 : next X : screen copy
physic to back : wait vbl : get sprite 0,0,I : cls
: next I
20530 for I=67 to 74 : sprite 1,100,100,I : screen
copy back to physic : screen copy physic to back :
for X=0 to 47 : gosub 20540 : next X : screen copy
physic to back : wait vbl : get sprite 0,0,I : cls
: next I
20531 cls : ink 3 : bar 0,0 to 3,3 : wait vbl : get
sprite 0,0,75
20535 goto 20750
20540 rem
20570 BLIT1$=chr$($A0)+chr$(0)+chr$($4E)+chr$($75)
20580 BLIT2$=chr$($A0)+chr$(7)+chr$($4E)+chr$($75)
20590 call varptr(BLIT1$)
20600 BLT_A$=space$(100)
20610 doke varptr(BLT_A$),1 : rem LARGEUR
20620 doke varptr(BLT_A$)+2,48 : rem HAUTEUR
20630 doke varptr(BLT_A$)+4,4 : doke varptr(BLT_A$)
+6,15
20640 doke varptr(BLT_A$)+8,0 : loke varptr(BLT_A$)
+10,$3030303
20650 doke varptr(BLT_A$)+14,X+100 : doke varptr
(BLT_A$)+16,100 : rem ORI(X/Y)
20660 loke varptr(BLT_A$)+18,physic : doke varptr
(BLT_A$)+22,8 : rem ORI(ECRAN)
20670 doke varptr(BLT_A$)+24,160 : doke varptr
(BLT_A$)+26,2
20680 doke varptr(BLT_A$)+28,47-X : doke varptr
(BLT_A$)+30,0 : rem SOURCE(X,Y)
20690 loke varptr(BLT_A$)+32,physic : doke varptr
(BLT_A$)+36,8 : rem DEST(ECR)
20700 doke varptr(BLT_A$)+38,160 : doke varptr
(BLT_A$)+40,2
20710 loke varptr(BLT_A$)+42,0
20720 areg(6)=varptr(BLT_A$)
20730 call varptr(BLIT2$)
20740 return
20750 cls : mode 1 : locate 0,12 : centre "Veuillez
insérer votre disk de travail"
20760 print : centre "(Je sauve la banque...)"
20770 wait key : save "A:MEDOR.MBS"
20780 cls : centre "Et bien jouez maintenant..."
20790 print : centre "Lancez le fichier HOTDOG.BAS"
: wait key : end
```

Le Basic STOS, c'est aussi sur
3615 STMAG
choix "*STO"

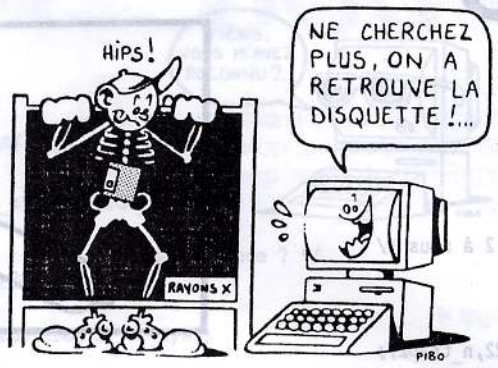
EDITEUR

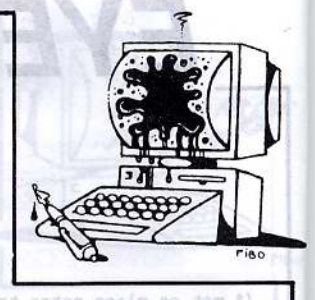
GFA-BASIC

Passez vos disquettes au peigne fin, grâce à cet éditeur de secteurs et à ses nombreuses options (recherche, modification, catalogue, recopie de secteurs, etc...).

Nicolas ROUGIER

UTILITAIRE
Moyenne Résolution






```
'
' EDITEUR
' Nicolas ROUGIER pour STMAGICIEL
'
IF XBIOS(4)<>1
  ALERT 3,"|          STOP    |Cet éditeur ne fonc
tionne |qu'en moyenne résolution",1," OK ",rep%
  END
ENDIF
init
debut
lirsec
affsec
PRINT CHR$(27);"p"
checksum
PRINT CHR$(27);"q"
menu
PROCEDURE menu
  DO
    key$=UPPER$(INKEY$)
    b=ASC(RIGHT$(key$))
    c=ASC(LEFT$(key$))
    cur=comp+cc
    IF b<>0 AND c=0
      IF b=72                    ! Monter
        IF cc>15
          SUB cc,16
        ELSE IF cur>255 AND cur<272
          page_haute
          cc=cc+240
        ENDIF
      ENDIF
      IF b=80                    ! Descendre
        IF cc<240
          ADD cc,16
        ELSE IF cur>239 AND cur<256
          page_basse
          cc=cc-240
        ENDIF
      ENDIF
      IF b=75                    ! Gauche
        IF cc>0
          SUB cc,1
        ELSE IF cur=256
          page_haute
          cc=255
        ENDIF
      ENDIF
      IF b=77                    ! Droite
        IF cc<255
          ADD cc,1
        ELSE IF cur=255
          page_basse
          cc=0
        ENDIF
      ENDIF
      affz
      GOTO loop
    ENDIF
    MOUSE x,y,k
    choix=INT((x-19)/56)+INT(y/8)*12
    IF k AND choix=0 OR key$="L"   ! Lire secteur
      PBOX 22,0,73,7
      info_fle
      key
      IF b=13
        lirsec
        affsec
        PRINT CHR$(27);"p"
        checksum
        PRINT CHR$(27);"q"
      ENDIF
      info_gen
      PBOX 22,0,73,7
    ENDIF
    IF k AND choix=12 OR key$="E"  ! Ecrire secteur
      PBOX 22,8,73,15
      info_fle
      key
      IF b=13
        ecrsec
      ENDIF
      info_gen
      PBOX 22,8,73,15
    ENDIF
    IF k AND choix=1 OR key$="T"   ! Texte
      PBOX 78,0,129,7
      text
      PBOX 78,0,129,7
    ENDIF
    IF k AND choix=13 OR key$="H"  ! Hexadécimal
      PBOX 78,8,129,15
      hexa
      PBOX 78,8,129,15
    ENDIF
    IF k AND choix=2 OR key$="Q"   ! Quit
      PBOX 134,0,185,7
      EDIT
    ENDIF
    IF k AND choix=14 OR key$="R"  ! Recherche
      PBOX 134,8,185,15
      rechdisk
      PBOX 134,8,185,15
    ENDIF
    IF k AND choix=3 OR key$="M"   ! Entrée buffer
      PBOX 190,0,241,7
      buffer$=tamp$
      PRINT CHR$(27);"p"
      PRINT AT(32,2);"plein"
      PRINT CHR$(27);"q"
      PBOX 190,0,241,7
    ENDIF
    IF k AND choix=15 OR key$="S"  ! Sortie buffer
      IF buffer$<>""
        PBOX 190,8,241,15
        PRINT CHR$(27);"p"
        PRINT AT(32,2);"vide "
        PRINT CHR$(27);"q"
        tamp$=buffer$
        buffer$=""
        affsec
        PBOX 190,8,241,15
      ENDIF
      affz
    ENDIF
    IF k AND choix=5 OR key$="&"   ! Page haute
      page_haute
    ENDIF
    IF k AND choix=17 OR key$="ê"  ! Page basse
      page_basse
    ENDIF
    IF k AND choix=18 OR key$="D"  ! Disque
      PBOX 358,8,409,15
      init
      PBOX 358,8,409,15
    ENDIF
    IF k AND choix=24 OR key$="K"  ! Catalogue
      PBOX 22,16,73,23
      catalog
      PBOX 22,16,73,23
    ENDIF
    IF k AND (choix=25 OR choix=26) OR key$="C"
    ' Checksum
      PBOX 78,16,185,23
      checksum
      PBOX 78,16,185,23
    ENDIF
    IF k AND choix=27 OR key$="B"  ! Boot
      PBOX 190,16,241,23
      boot
      PBOX 190,16,241,23
    ENDIF
    loop:
  LOOP
RETURN
PROCEDURE debut
  CLS
  DIM cat$(200),rp$(200)
  sec%=1
  pis%=0
  face%=0
  SETCOLOR 0,&H777
  SETCOLOR 1,&H70
  SETCOLOR 2,&H700
  SETCOLOR 3,&H0
  BOUNDARY 0
  GRAPHMODE 3
  DEFTEXT 1,0,0,6
  TEXT 0,6,CHR$(14)+CHR$(15)+" L.sect  Text   Quit
M.buff Buffer  Pg "+CHR$(1)+"  Face 1 Sec 01 Octet
Hex :"
  TEXT 0,14,"  E.sect  Hexa   Rech  S.buff  vide
Pg "+CHR$(2)+"  Disque Pis 00 N° :   Dec :"
  TEXT 0,22,"   Cata                  Boot
Bin :"
  PBOX 0,0,641,23
  FOR i=19 TO 570 STEP 56
    LINE i,0,i,23
    LINE i+1,0,i+1,23
  NEXT i
  LINE 19+2*56,16,19+2*56,23
  LINE 19+2*56+1,16,19+2*56+1,23
  PRINT CHR$(27);"q"
  info_gen
RETURN
PROCEDURE aff
  PRINT CHR$(27);"p"
  PRINT AT(57,1);RIGHT$("00"+STR$(sec%),2)
  PRINT AT(57,2);RIGHT$("00"+STR$(pis%),2)
  PRINT AT(51,1);STR$(face%+1)
  PRINT CHR$(27);"q"
RETURN
PROCEDURE init
  tst%=XBIOS(8,L:HIMEM,L:0,W:0,W:1,W:0,W:0,W:1)
  IF tst%
    trait_erreur
  ENDIF
  nbsect=PEEK(HIMEM+24)
  nbface=PEEK(HIMEM+26)
  nbpis=(PEEK(HIMEM+20)*256+PEEK(HIMEM+19))
/nbface/nbsect-1
RETURN
PROCEDURE ecrsec
  adr%=VARPTR(tamp$)
  tst%=XBIOS(9,L:adr%,L:1,0,sec%,pis%,face%,1)
  IF tst%
    trait_erreur
  ENDIF
  info_gen
RETURN
PROCEDURE lirsec
  tamp$=SPACE$(512)
  adr%=VARPTR(tamp$)
  tst%=XBIOS(8,L:adr%,L:1,0,sec%,pis%,face%,1)
  IF tst%
    trait_erreur
  ENDIF
  info_gen
RETURN
PROCEDURE affsec
  comp=MUL(page,256)
  adr%=V:tamp$
  FOR i=5 TO 20
    tt$=HEX$(comp,3)+": "
    PRINT AT(61,i);
    FOR j=1 TO 16
      tt$=tt$+RIGHT$(HEX$(PEEK(ADD(adr%,comp)),
2),2)+" "
      OUT 5,PEEK(ADD(adr%,comp))
      INC comp
    NEXT j
    PRINT AT(1,i);tt$
  NEXT i
  comp=MUL(page,256)
  PRINT AT(xx+61,yy+5);
  OUT 5,PEEK(adr%+cc+comp)
  affz
RETURN
PROCEDURE rechdisk
  CLR find%
  GRAPHMODE 1
  DEFFILL 0
  PBOX 0,24,639,199
  ALERT 2,"| |Type de recherche|",1," HEXA
TEXT ",rep%
  IF rep%=2
    PRINT AT(1,4);"Quel est le texte à rechercher : ";
    FORM INPUT 10,tt$
  ELSE
    CLR tt$
    PRINT AT(1,4);"Séquence à rechercher : ";
    FORM INPUT 20,hh$
    hh$=UPPER$(hh$)
    IF ODD(LEN(hh$))
      hh$="0"+hh$
    ENDIF
    FOR i=1 TO LEN(hh$)-1 STEP 2
      tt$=tt$+CHR$(VAL("&H"+MID$(hh$,i,2)))
    NEXT i
  ENDIF
  IF tt$=""
    GOTO fin
  ENDIF
  hauteur=50
  DEFTEXT 1,0,0,4
  TEXT 60,hauteur,"000000000011111111112222222222
333333333344444444455555555556666666666777777777
888888     x10" ! 5 espaces à la fin
  TEXT 60,hauteur+5,"01234567890123456789012345678 9
012345678901234567890123456789012345678901234567890
12345     x1" ! itou, 5 espaces...
  TEXT 1,62,"FACE 1"
  TEXT 1,122,"FACE 2"
  FOR i=1 TO 9
    TEXT 50,56+i*6,STR$(i)
    TEXT 50,116+i*6,STR$(i)
  NEXT i
  PRINT CHR$(27);"p"
  DEFFILL 1
  FOR trkk%=pis% TO nbpis
    FOR sidd%=face% TO nbface-1
      key$=INKEY$
      tamp$=SPACE$(512*nbsect)
      adr%=V:tamp$
      tst%=XBIOS(8,L:adr%,L:1,0,1,trkk%,sidd%,
nbsect)
      IF tst%
        trait_erreur
      ENDIF
      FOR i=0 TO 8
        PSET trkk%*6+60,62+sidd%*60+i*6,3
      NEXT i
      PRINT AT(57,2);RIGHT$("00"+STR$(trkk%),2)
      PRINT AT(51,1);STR$(sidd%+1)
      tst%=INSTR(tamp$,tt$)
      IF tst%
        i=INT(tst%/512)
        PBOX trkk%*6+59,60+sidd%*60+i*6,trkk%*6+63,
62+sidd%*60+i*6
        sec%=i+1
        pit%=trkk%
        fa%=sidd%
        ret%=tst%
        find%=1
      ENDIF
      EXIT IF key$=CHR$(27)
    NEXT sidd%
    EXIT IF key$=CHR$(27)
  NEXT trkk%
  IF key$=CHR$(27) OR find%=0
    GOTO fin
  ENDIF
  PRINT CHR$(27);"q"
  pis%=pit%
  face%=fa%
  lirsec
  ret%=ret%-INT(ret%/512)*512
  IF ret%>255
    cc=ret%-257
    page=1
  ELSE
    cc=ret%-1
    page=0
  ENDIF
  fin:
  DEFFILL 0
  PBOX 0,24,639,170
  DEFTEXT ,,,6
  GRAPHMODE 3
  lirsec
  affsec
  affsec
  info_gen
RETURN
PROCEDURE affz
  yy=INT(cc/16)
  xx=cc-(yy*16)
  a=PEEK(V:tamp$+cc+comp)
  GRAPHMODE 2
  DEFFILL 2
  PBOX (xx+60)*8,(yy+4)*8,(xx+61)*8-1,(yy+5)*8-1
  DEFTEXT 3
  TEXT (xx+60)*8,(yy+5)*8-2,CHR$(a)
  PRINT CHR$(27);"p"
  PRINT AT(72,1);HEX$(a,2)
  PRINT AT(72,3);BIN$(a,8)
  PRINT AT(72,2);RIGHT$("000"+STR$(a),3)
  PRINT AT(63,2);HEX$(cc+comp,3)
  PRINT CHR$(27);"q"
  IF cc1<>cc
    PRINT AT(xx1+61,yy1+5);
    OUT 5,PEEK(VARPTR(tamp$)+cc1+comp)
    PRINT AT(5+xx1*3,yy1+5);" ";HEX$(PEEK(V:tamp$
+cc1+comp),2)+" " ! un espace entre guillemets
  ENDIF
  TEXT (4+xx*3)*8,(yy+5)*8-2,"[  ]"  ! 2 espaces
  DEFTEXT 2
  TEXT (5+xx*3)*8,(yy+5)*8-2,HEX$(a,2)
  yy1=yy
  xx1=xx
  cc1=cc
  GRAPHMODE 3
RETURN
PROCEDURE text
  REPEAT
    key$=INKEY$
    IF key$<>"" AND cc<255 AND key$<>CHR$(13)
      yy=INT(cc/16)
      xx=cc-(yy*16)
      PRINT AT(xx+61,yy+5);
      OUT 5,ASC(key$)
      POKE (V:tamp$+cc+comp),ASC(key$)
      PRINT AT(6+xx*3,yy+5);HEX$(ASC(key$),2)
      INC cc
      affz
      cc1=cc
    ENDIF
  UNTIL key$=CHR$(13)
  PRINT CHR$(27);"p"
  checksum
  PRINT CHR$(27);"q"
RETURN
PROCEDURE key
  REPEAT
    key$=INKEY$
    IF UPPER$(key$)="F"
      INC face%
      IF face%>=nbface
        face%=0
      ENDIF
      aff
    ENDIF
    b=ASC(RIGHT$(key$))
    c=ASC(LEFT$(key$))
    IF b<>0 AND c=0
      IF b=80          !Monter
        INC sec%
        IF sec%>nbsect
          sec%=nbsect
        ENDIF
      ENDIF
      IF b=72          !Descendre
        DEC sec%
        IF sec%<1
          sec%=1
        ENDIF
      ENDIF
      IF b=77          !Gauche
        INC pis%
        IF pis%>nbpis
          pis%=nbpis
        ENDIF
      ENDIF
      IF b=75          !Droite
        DEC pis%
        IF pis%<0
          pis%=0
        ENDIF
      ENDIF
      aff
    ENDIF
  UNTIL b=13 OR b=27
RETURN
PROCEDURE page_basse
  PBOX 302,8,353,15
  page=1
  affsec
  PBOX 302,8,353,15
RETURN
PROCEDURE page_haute
  PBOX 302,0,353,7
  page=0
```

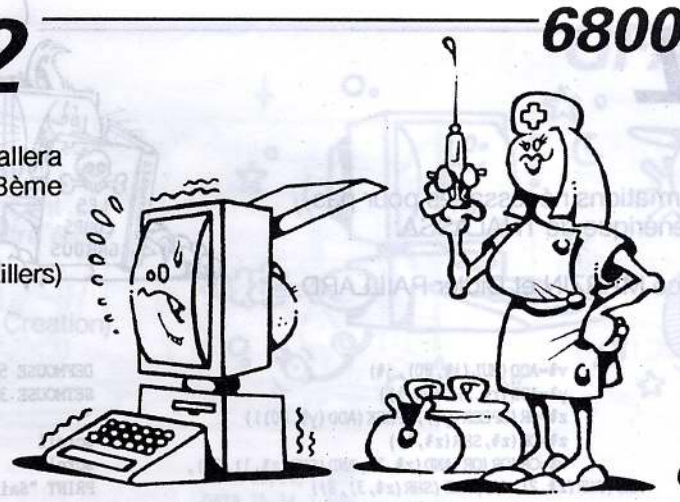
PROBLEMES EN GFA ?
3615 STMAG: *GFA.

suite en page 9

VIRUS KICKER 2

Voilà de quoi soigner vos disquettes (pas vos jeux originaux!). On installera le programme dans le boot-secteur d'une disquette vierge, grâce au 3ème listing. Pas trop recommandé aux novices.

JEDI & ULTIMATUM (from The Heavy Killers)



PiBo

ANTI-VIRUS
Toutes Résolutions

LE LISTING ASSEMBLEUR

```
*          Viruskicker 2
* par: - Jedi of Sector One
*      - Ultimatum of Zap Creation
*      from the Heavy Killers
*          Viruskicker 2
* par: - Jedi of Sector One
*      - Ultimatum of Zap Creation
*  (from the Heavy Killers)
start   pea thk(pc)
        move #9,-(sp)
        trap #1
        lea 2(sp),a5
        lea pat(pc),a0
        move.l $42e.w,d0
check   movea.l d0,a6
        cmpi.l #$12123456,(a6)
        beq.s virus
        subi.l #$200,d0
        bne.s check
        lea ok(pc),a0
virus   move.l a0,(a5)
        trap #1
        lea menu(pc),a0
        move.l a0,(a5)
        trap #1
        subq.w #2,(sp)
key     trap #1
        swap d0
        cmpi.b #$3b,d0
        bne.s nan1
        move #$2700,sr
        lea 8.w,a1
        lea rstrout(pc),a0
        moveq #9,d7
        move.l (a0)+,(a1)+
        dbf d7,*-2
        jmp $12.w
rstrout dc.l 12
        movea.l 4.w,a0
        jmp (a0)
        lea nan1(pc),a0
        moveq #0,d0
        move.l d0,d1
        move.l d1,d2
        move.l d2,d3
        move.l d3,d4
        move.l d4,d5
        move.l d5,d6
        move.l d6,d7
killall movem.l d0-d7,(a0)
        add #$20,a0
        bra.s killall
nan1    cmpi.b #$3e,d0
        bne.s nan2
        bchg #1,$ffff820a.w
nan2    cmpi.b #$39,d0
        bne.s nan3
        addq.l #6,sp
        rts
nan3    cmpi.b #$3d,d0
        bne.s nan4
        lea csr(pc),a6
        move.l a6,(a5)
        addq.w #2,(sp)
        trap #1
        lea label(pc),a6
        move #$1a00,(a6)
        move.l a6,(a5)
        addq.w #1,(sp)
        trap #1
        move #$2020,(a6)
        addq.l #6,sp
youpi   bra start
nan4    cmpi.b #$3c,d0
rtn     bne key
        movea.l $44e.w,a6
        move #1,(sp)
        clr.l -(sp)
        pea 1.w
        clr.l -(sp)
        pea 32000(a6)
        move #8,-(sp)
        trap #14
        movem.l 32008(a6),d1-d5
        lea start-28(pc),a6
        movem.l d1-d5,8(a6)
        move #9,(sp)
        move.l a6,2(sp)
        move #254,d1
        add (a6)+,d0
        dbf d1,*-2
        move #$1234,d1
        sub d0,d1
        move d1,(a6)
        trap #14
        lea 24(sp),sp
        bra.s youpi
csr     dc.b 27,'e',0
thk     dc.b 27,'f',27,'E',27,'Y'
        dc.b 32,42,'Jedi/Sector One/THK'
        dc.b 27,'Y',34,48,'presents'
        dc.b 27,'Y',36,45,'Viruskicker 2',0
pat     dc.b 27,'Y',40,45,'*VIRUS ALERT*',7,0
ok      dc.b 27,'Y',40,47,'-No virus-',0
menu    dc.b 27,'Y',44,32,'Disk name:'
label   dcb.b 28,32
        dc.b 13,10,10,'F1:Reset',13,10
        dc.b 'F2:Copy dis boot',13,10
        dc.b 'F3:Set name',13,10
        dc.b 'F4:Sync',13,10
        dc.b 'Space:Quit'
        dc.b 27,'Y',44,44,0
```

```
' Ce listing génère directement le fichier
' VRSKICKR.PRG
thk$=STRING$(28,0)
FOR g%=0 TO 120
  READ zc$
  thk$=thk$+MKL$(VAL("&H"+zc$))
NEXT g%
BSAVE "VRSKICKR.PRG",VARPTR(thk$),512
DATA 487A0117,3F3C0009,4E414BEF,241FA,1422038
DATA 42E2C40,C961212,3456670C,4800000,20066EE
DATA 41FA013B,2A884E41,41FA0142,2A884E41,55574E41
DATA 48400C00,3B6640,46FC2700,43F80008,41FA000E
DATA 7E0922D8,51CFFFFC,4EF80012,C,20780004
DATA 4ED041FA,1C7000,22002401,26022803,2A042C05
DATA 2E0648D0,FFD0FC,2060F6,C00003E,66060878,1820A
DATA C000039,66045C8F,4E750C00,3D6622,4DFA0074
DATA 2A8E5457,4E414DFA,D63CBC,1A002A8E,52574E41
DATA 3CBC2020,5C8F6000,FF400C00,3C6600,FF722C78,44E3EBC
DATA 142A7,48780001,42A7486E,7D003F3C,84E4E,4CEE003E
DATA 7D084DFA,FEF848EE,3E0008,3EBC0009,2F4E0002
DATA 323C00FE, D05E51C9,FFFC323C,12349240,3C814E4E
DATA 4FEF0018,60A81B65,1B661B,451B5920,2A4A6564
DATA 692F5365,63746F72,204F6E65,2F54484B,1B592230
DATA 70726573,656E7473,1B59242D,56697275,736B6963
DATA 6B657220,32001B59,282D2A56,49525553,20414C45
DATA 52542A07,1B5928,2F2D4E6F,20766972,75732D00
DATA 1B592C20,4469736B,206E616D,653A2020,20202020
DATA 20202020,20202020,20202020,20202020,20202020
DATA 20200D0A,A46313A,52657365,740D0A46,323A436F
DATA 70792064,69732062,6F6F740D,A46333A,53657420
DATA 6E616D65,D0A4634,3A53796E,630D0A53
DATA 70616365,3A517569,741B592C,2C000000
```

LE LISTING GFA POUR INSTALLER
LE PROGRAMME EN BOOT-SECTEUR

```
OPEN "i",#1,"VRSKICKR.PRG"
SEEK #1,28
a$=INPUT$(484,#1)
CLOSE #1
thk$=STRING$(1024,0)
ad%=VARPTR(thk$) AND &HFFFFFFFE
~XBIOS(8,L:ad%,L:0,0,1,0,0,1)
BMOVE VARPTR(a$),ad%+28,512
LPOKE ad%,&H31FC0765
LPOKE ad%+4,&H82406014
FOR a%=ad% TO ad%+508 STEP 2
  ADD chk%,DPEEK(a%)
NEXT a%
DPOKE ad%+510,&H1234-chk%
~XBIOS(9,L:ad%,L:0,0,1,0,0,1)
```

EDITEUR

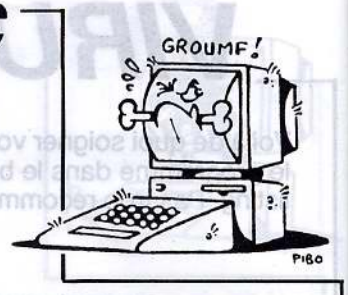
suite de la page 8

```
  affsec
  PBOX 302,0,353,7
RETURN
PROCEDURE hexa
  DO
    GRAPHMODE 1
    comp=page*256
    yy=INT(cc/16)
    xx=cc-(yy*16)
    tt$=MID$(HEX$(PEEK(V:tamp$+cc+comp),2),1,1)
    DEFTEXT 3
    TEXT (4+xx*3)*8,(yy+5)*8-2,"[   ]" ! 2 espaces
    DEFTEXT 2
    TEXT (5+xx*3)*8,(yy+5)*8-2,HEX$(PEEK(V:tamp$
+cc+comp),2)
    DEFFILL 2
    GRAPHMODE 1
    PBOX (xx+60)*8,(yy+4)*8,(xx+61)*8-1,(yy+5)*8-1
    GRAPHMODE 2
    DEFTEXT 3
    TEXT (xx+60)*8,(yy+5)*8-2,CHR$(PEEK(V:tamp$
+cc+comp))
    REPEAT
      PRINT CHR$(27);"p"
      PRINT AT(63,2);HEX$(cc+comp,3)
      PRINT CHR$(27);"q"
      REPEAT
        key$=UPPER$(INKEY$)
        EXIT IF key$=CHR$(32) OR key$=CHR$(13)
      UNTIL INSTR("0123456789ABCDEF",key$)
      a=VAL("&H"+key$+tt$)
      tt$=key$
      IF key$=CHR$(32) OR key$=CHR$(13)
        a=PEEK(V:tamp$+cc+comp)
      ENDIF
      EXIT IF key$=CHR$(13)
      PRINT CHR$(27);"p"
      PRINT AT(72,1);HEX$(a,2)
      PRINT AT(72,2);RIGHT$("000"+STR$(a),3)
      PRINT AT(72,3);BIN$(a,8)
      PRINT CHR$(27);"q"
      GRAPHMODE 1
      DEFTEXT 2
      TEXT (5+xx*3)*8,(yy+5)*8-2,HEX$(a,2)
      DEFFILL 2
      PBOX (xx+60)*8,(yy+4)*8,(xx+61)*8-1,(yy+5)*8-1
      GRAPHMODE 2
      DEFTEXT 3
      TEXT (xx+60)*8,(yy+5)*8-2,CHR$(a)
      POKE V:tamp$+cc+comp,a
    UNTIL key$=CHR$(32)
    PRINT AT(5+xx*3,yy+5);" ";HEX$(a,2);" "
    PRINT AT(61+xx,yy+5);
    OUT 5,a
    EXIT IF key$=CHR$(13)
    INC cc
    IF cc=256 AND page=0
      cc=0
      GRAPHMODE 3
      page_basse
    ENDIF
    IF cc>255 AND page=1
      cc=255
    ENDIF
  LOOP
  affz
  PRINT CHR$(27);"p"
  checksum
  PRINT CHR$(27);"q"
RETURN
PROCEDURE trait_erreur
  er$=MID$(ERR$(tst%),5)
  er$=MID$(er$,1,LEN(er$)-9)+"  "
  er$=MID$(er$,SUCC(2*ABS(MID$(er$,1,1)="*")))
  IF INSTR(er$,"|")
    er$=MID$(er$,1,INSTR(er$,"|")-1)+"  |"+MID$(er$
,SUCC(INSTR(er$,"|")))
  ENDIF
  ALERT 0,er$,1,"  OK  ",tst%
RETURN
PROCEDURE catalog
  n_disk$=" AUCUN"
  CLR nbf,nbd,c
  detail_directory
  GRAPHMODE 1
  DEFFILL 0
  PBOX 0,24,639,199
  COLOR 1
  BOX 0,28,639,122
  BOX 0,122,380,198
  BOX 380,122,639,198
  PRINT AT(2,17);"FICHIER  :"
  PRINT AT(2,18);"LONGUEUR :"      ! Arrangez-vous
  PRINT AT(2,20);"TYPE     :"      ! pour que ce
  PRINT AT(2,21);"GENRE    :"      ! soit joli...
  PRINT AT(2,22);"ECRITURE :"      ! Vous êtes
  PRINT AT(2,23);"OBJET    :"      ! assez grands.
  PRINT AT(50,17);"NOM DU DISQUE:";n_disk$
  PRINT AT(50,18);"NOMBRE DE FACES: ";nbface
  PRINT AT(51,19);SUCC(nbpis);" PISTES PAR FACE"
  PRINT AT(52,20);nbsect;" SECTEURS PAR PISTE"
  FOR i=1 TO nbf
    ADD c,CVL(RIGHT$(rp$(i),4))
  NEXT i
  PRINT AT(50,21);"PLACE OCCUPEE: ";c;" Octets"
  PRINT AT(50,22);"PLACE LIBRE  : ";DFREE(0);" Octect"
  PRINT AT(50,23);"NOMBRE DE DOSSIERS: ";nbd
  PRINT AT(50,24);"NOMBRE D'OBJETS: ";nbf-nbd
  FOR i=1 TO ABS(nbf<12)*nbf+ABS(nbf>12)*11
    PRINT AT(4,i+4);MID$(cat$(i),4)
  NEXT i
  cur=1
  cy=1
  DO
    PRINT AT(2,cy+4);"< "+MID$(cat$(cur),4)+" >"+
SPACE$(30)
    PRINT AT(13,17);LEFT$(MID$(cat$(cur),SUCC
(RINSTR(cat$(cur),"\")))+SPACE$(30),30)
    PRINT AT(13,18);CVL(RIGHT$(rp$(cur),4));
" Octets";SPACE$(8)
    res$=BIN$(ASC(MID$(rp$(cur),1,1)),5)
    PRINT AT(16,20);MID$("NormalSystem",VAL(MID$
(res$,3,1))*7,6)
    PRINT AT(16,21);MID$("FichierDossier",VAL(MID$
(res$,1,1))*8,7)
    PRINT AT(16,22);MID$("Read-WriteRead only ",VAL
(MID$(res$,5,1))*11,10)
    PRINT AT(16,23);MID$("Non cacheCache     ",VAL
(MID$(res$,4,1))*9+1,9)
    key$=INKEY$
    b=ASC(RIGHT$(key$))
    c=ASC(LEFT$(key$))
    IF (b=80 OR b=72) AND c=0
      PRINT AT(2,cy+4);"  "+MID$(cat$(cur),4)+
SPACE$(30)
    ENDIF
    IF b=80 AND c=0 AND cur<nbf   ! Descendre
      INC cur
      IF cy<11
        INC cy
      ELSE
        BMOVE XBIOS(3)+40*160,XBIOS(3)+32*160,
80*160
      ENDIF
    ENDIF
    IF b=72 AND c=0 AND cur>1  ! Monter
      DEC cur
      IF cy>1
        DEC cy
      ELSE
        BMOVE XBIOS(3)+32*160,XBIOS(3)+40*160,
80*160
      ENDIF
    ENDIF
    EXIT IF key$=CHR$(13) OR key$=CHR$(27)
  LOOP
  PBOX 0,24,639,199
  affsec
  info_gen
RETURN
PROCEDURE detail_directory
  tst%=FSFIRST("A:\"+path$+"*.*",-1)
  DO UNTIL tst%
    IF LEFT$(CHAR{FGETDTA()+30})<>"."
      IF BYTE{FGETDTA()+21} AND 8
        n_disk$=LEFT$(CHAR{FGETDTA()+30},60)
      ELSE
        INC nbf
        cat$(nbf)=LEFT$("A:\"+path$+CHAR{
FGETDTA()+30},60)
        rp$(nbf)=CHR$(BYTE{FGETDTA()+21})+MKL$
({FGETDTA()+26})
      ENDIF
      IF BYTE{FGETDTA()+21} AND 16
        INC nbd
        path$=path$+CHAR{FGETDTA()+30}+"\"
        detail_directory
        tst%=FSFIRST("A:\"+LEFT$(path$,RINSTR
(path$,"\",LEN(path$)-1))+"*.*",-1)
        WHILE path$<>LEFT$(path$,RINSTR(path$,"\",
LEN(path$)-1))+CHAR{FGETDTA()+30}+"\"
          tst%=FSNEXT()
        WEND
        path$=LEFT$(path$,RINSTR(path$,"\",LEN
(path$)-1))
      ENDIF
    ENDIF
    tst%=FSNEXT()
  LOOP
RETURN
PROCEDURE checksum
  CLR check
  FOR i=V:tamp$ TO V:tamp$+511 STEP 2
    ADD check,DPEEK(i)
  NEXT i
  PRINT AT(11,3);"Check : ";HEX$(check,4);" "
RETURN
PROCEDURE boot
  stamp$=tamp$
  sec%=1
  pis%=0
  face%=0
  lirsec
  CLR check
  FOR i=V:tamp$ TO V:tamp$+511 STEP 2
    ADD check,DPEEK(i)
  NEXT i
  IF HEX$(check,4)="1234" AND DPEEK(V:tamp$)<>0
    ALERT 2,"| | BOOT EXECUTABLE |PEUT-ETRE UN VI
RUS ",1," OK  ",rep%
  ELSE
    ALERT 0,"PAS DE BOOT EXECUTABLE  ",1,
" OK  ",rep%
  ENDIF
  tamp$=stamp$
RETURN
PROCEDURE info_gen
  PRINT AT(1,22);"[L]:Lire secteur [T]:Texte
[Q]:Quitter [M]:Mettre dans buffer [S]:Sortie buf-
fer"
  PRINT AT(1,23);"[E]:Ecrire secteur [H]:hexa
[R]:Recherche [D]:Disque [K]:Catalogue [B]:Boot"
  PRINT AT(1,24);"[C]:Checksum. Utiliser ";
  OUT 5,1
  OUT 5,2
  OUT 5,3
  OUT 5,4
  PRINT " pour déplacer le curseur.";
RETURN
PROCEDURE info_fle
  PRINT AT(1,22);SPC(80)
  PRINT AT(1,23);"        Utiliser ";
  OUT 5,1
  OUT 5,2
  PRINT " pour choisir le secteur, ";
  OUT 5,3
  OUT 5,4
  PRINT " pour choisir la piste,       ";
  PRINT AT(1,24);SPC(80);
  PRINT AT(1,24);"                [F] pour choisir la
face, [Esc] pour sortir " ! 14 espaces au début
  PRINT AT(1,1)
RETURN
```

Minitel
STMAGICIEL:
3615 STMAG

INTERPOL

GFA-BASIC



Un utilitaire qui calcule et dessine les transformations nécessaires pour passer d'une image à une autre, à la façon du générique de THALASSA.

Laurence MARZIN et Didier PAILLARD

```
' ********************************
' *  Interpol, (C)1990 D.Paillard *
' ********************************
init
DO
  ON MENU
LOOP
PROCEDURE init
  RESERVE 200000
  '
  max%=100
  DIM m$(26),x%(max%),y%(max%),lig%(max%,3),tab%
(max%,7)
  '
  image$=SPACE$(32066)
  pimage$=STRING$(16000,0)
  '
  RESTORE
  FOR i%=0 TO 26
    READ m$(i%)
  NEXT i%
  MENU m$()
  ON MENU GOSUB evaluation
  '
  OPENW 0
  scroll
  PRINT "Bienvenue dans INTERPOL"
  '
  raz
  refresh
  '
  DATA Bureau,  Information,————————,1,2,3,4,5,6,
  DATA Fichier,- Calque —-,  Charger,  Effacer,
  DATA - Animation -,  Charger,  Sauver,————,
  DATA  Quitter,Edition,  Points  ,  Lignes  ,
  DATA   Effacer,Animation, Démarrer  ,
RETURN
PROCEDURE quitte
  scroll
  scroll
  PRINT "Fin du programme ?"
  '
  ALERT 3,"Voulez-vous quitter|le
programme ?",2,"Oui|Non",rep%
  IF rep%=1
    END
  ENDIF
RETURN
PROCEDURE scroll
  BMOVE XBIOS(2)+2960,XBIOS(2)+1680,12800
  PRINT AT(1,11);SPACE$(80);
  PRINT AT(1,11);
RETURN
PROCEDURE evaluation
  MENU OFF
  m%=MENU(0)
  SELECT m%
  CASE 1
    ALERT 0,"   Interpolations| (c) février 1990|
L.Marzin & D.Paillard  ",1," Ok ",dummy%
  CASE 12
    @load_calque
  CASE 13
    @efface_calque
  CASE 15
    @load_animat
  CASE 16
    @save_animat
  CASE 18
    @quitte
  CASE 21
    @saisie_pt
  CASE 22
    @saisie_lig
  CASE 23
    @raz
    @refresh
  CASE 26
    @animate
  ENDSELECT
RETURN
PROCEDURE raz
  FOR i%=0 TO max%
    x%(i%)=-1
    lig%(i%,0)=-1
  NEXT i%
RETURN
PROCEDURE load_calque
  LOCAL nom$,c%,i%,j%,v%,y%,z%,w%,r%
  '
  scroll
  scroll
  PRINT "Chargement d'un calque"
  '
  DO
    FILESELECT "\*.pi3","",nom$
    IF RIGHT$(nom$)<>"\"
      c%=0
    ENDIF
  LOOP UNTIL c%=0
  '
  IF nom$<>""
    scroll
    PRINT "Nom du calque : ";nom$
    '
    winselect
    BLOAD nom$,V:image$
    scroll
    PRINT "Patientez environ 30 secondes. Réduction
de l'image"
    '
    r%=ADD(V:image$,34)
    FOR i%=0 TO 199
      FOR j%=0 TO 39 STEP 2
        v%=ADD(MUL(i%,80),j%)
        y%=ADD(r%,SHL(v%,1))
        z%=OR(LPEEK(y%),LPEEK(ADD(y%,80)))
        z%=OR(z%,SHR(z%,1))
        w%=OR(OR(OR(AND(z%,1),AND(SHR(z%,1),2)),
AND(SHR(z%,2),4)),AND(SHR(z%,3),8))
        w%=OR(OR(OR(w%,AND(SHR(z%,4),16)),AND(SHR
(z%,5),32)),AND(SHR(z%,6),64))
        w%=OR(OR(OR(w%,AND(SHR(z%,7),128)),AND(SHR
(z%,8),256)),AND(SHR(z%,9),512))
        w%=OR(OR(OR(w%,AND(SHR(z%,10),1024)),AND
(SHR(z%,11),2048)),AND(SHR(z%,12),4096))
        w%=OR(OR(OR(w%,AND(SHR(z%,13),8192)),AND
(SHR(z%,14),16384)),AND(SHR(z%,15),32768))
        DPOKE dest%,w%
        ADD dest%,2
      NEXT j%
      ADD dest%,40
    NEXT i%
    refresh
  ELSE
    scroll
    PRINT "Pas de chargement"
  ENDIF
RETURN
PROCEDURE efface_calque
  scroll
  scroll
  PRINT "Effacement d'un calque"
  '
  winselect
  FOR i%=0 TO 199
    FOR j%=0 TO 39 STEP 4
      LPOKE dest%,0
      ADD dest%,4
    NEXT j%
    ADD dest%,40
  NEXT i%
  refresh
RETURN
PROCEDURE load_animat
  LOCAL c%,i%,j%,nom$
  c%=-1
  '
  scroll
  scroll
  PRINT "Chargement d'un fichier animation"
  '
  DO
    FILESELECT "\*.ANA","",nom$
    IF RIGHT$(nom$)<>"\"
      c%=0
    ENDIF
  LOOP UNTIL c%=0
  '
  IF nom$<>""
    scroll
    PRINT "Sous le nom : ";nom$
    '
    OPEN "i",#1,nom$
    FOR i%=0 TO max%
      INPUT #1,x%(i%)
      INPUT #1,y%(i%)
    NEXT i%
    FOR i%=0 TO max%
      INPUT #1,lig%(i%,0)
      INPUT #1,lig%(i%,1)
      INPUT #1,lig%(i%,2)
      INPUT #1,lig%(i%,3)
    NEXT i%
    CLOSE #1
    refresh
  ELSE
    scroll
    PRINT "Pas de chargement"
  ENDIF
RETURN
PROCEDURE save_animat
  LOCAL c%,i%,nom$
  '
  scroll
  scroll
  PRINT "Sauvegarde d'un fichier d'animation"
  '
  c%=-1
  DO
    FILESELECT "\*.ANA","",nom$
    IF RIGHT$(nom$)<>"\"
      c%=0
    ENDIF
  LOOP UNTIL c%=0
  '
  IF nom$<>""
    scroll
    PRINT "Fichier sauvé sous le nom : ";nom$
    '
    OPEN "o",#1,nom$
    FOR i%=0 TO max%
      PRINT #1,x%(i%)
      PRINT #1,y%(i%)
    NEXT i%
    FOR i%=0 TO max%
      PRINT #1,lig%(i%,0)
      PRINT #1,lig%(i%,1)
      PRINT #1,lig%(i%,2)
      PRINT #1,lig%(i%,3)
    NEXT i%
    CLOSE #1
  ELSE
    scroll
    PRINT "Pas de sauvegarde"
  ENDIF
RETURN
PROCEDURE saisie_pt
  DEFMOUSE 5
  SETMOUSE 319,299
  '
  scroll
  scroll
  PRINT "Saisie des points :"
  scroll
  PRINT "Cliquez pour saisir un nouveau point, ou
recliquez sur un ancien point,"
  scroll
  PRINT "pour l'effacer (efface aussi les lignes
concernées)."
  scroll
  scroll
  PRINT "Cliquez sur le bouton droit pour sortir."
  '
  GRAPHMODE 3
  DO
    tst_mouse
    IF mk%=1
      mxs%=mx%-3
      mxi%=mx%+3
      mys%=my%-3
      myi%=my%+3
      m%=-1
      i%=0
      DO
        IF x%(i%)>mxs% AND x%(i%)<mxi% AND y%(i%)
>mys% AND y%(i%)<myi%
          m%=i%
        ENDIF
        INC i%
      LOOP UNTIL m%<>-1 OR i%=max%
      IF m%=-1
        c%=-1
        i%=0
        DO
          IF x%(i%)=-1
            x%(i%)=mx%
            y%(i%)=my%
            c%=0
            PLOT mxs%,mys%
            BOX mxs%,mys%,mxi%,myi%
          ENDIF
          INC i%
        LOOP UNTIL c%=0 OR i%=max%
      ELSE
        PLOT x%(m%)-3,y%(m%)-3
        BOX x%(m%)-3,y%(m%)-3,x%(m%)+3,y%(m%)+3
        FOR j%=0 TO max%
          IF lig%(j%,0)<>-1
            IF lig%(j%,0)=m% OR lig%(j%,1)=m% OR
lig%(j%,2)=m% OR lig%(j%,3)=m%
              LINE x%(lig%(j%,0)),y%(lig%(j%,0)),x%
(lig%(j%,1)),y%(lig%(j%,1))
              LINE x%(lig%(j%,2)),y%(lig%(j%,2)),x%
(lig%(j%,3)),y%(lig%(j%,3))
              lig%(j%,0)=-1
            ENDIF
          ENDIF
        NEXT j%
        x%(m%)=-1
      ENDIF
      DO
        tst_mouse
      LOOP UNTIL mk%=0
    ENDIF
  LOOP UNTIL MOUSEK=2
  GRAPHMODE 1
  DEFMOUSE 0
  '
  scroll
  PRINT "Fin d'édition de points"
RETURN
PROCEDURE saisie_lig
  DEFMOUSE 5
  '
  scroll
  scroll
  PRINT "Saisie d'une ligne :"
  scroll
  PRINT "Cliquez successivement sur deux points
dans la fenêtre de départ"
  scroll
  PRINT "pour définir la ligne de départ, puis sur
deux points dans la fenêtre"
  scroll
  PRINT "d'arrivée pour la ligne d'arrivée. Et
ainsi de suite..."
  scroll
  scroll
  PRINT "Cliquez sur le bouton droit pour sortir."
  '
  GRAPHMODE 3
  DO
    tst_mouse
    IF mk%=1
      m%=-1
      i%=0
      DO
        IF x%(i%)<320 AND x%(i%)>mx%-6 AND
x%(i%)<mx%+6 AND y%(i%)>my%-6 AND y%(i%)<my%+6
          m%=i%
          dep1%=m%
        ENDIF
        INC i%
      LOOP UNTIL m%<>-1 OR i%=max%
      IF m%<>-1
        DO
        LOOP UNTIL MOUSEK=0
        smx%=x%(m%)
        smy%=y%(m%)
        DO
          DO
            tst_mouse
            LINE x%(m%),y%(m%),smx%,smy%
            LINE x%(m%),y%(m%),mx%,my%
            smx%=mx%
            smy%=my%
          LOOP UNTIL mk%=1
          n%=-1
          j%=0
          DO
            IF x%(j%)<320 AND x%(j%)>mx%-6 AND
x%(j%)<mx%+6 AND y%(j%)>my%-6 AND y%(j%)<my%+6
              n%=j%
              arr1%=n%
            ENDIF
            INC j%
          LOOP UNTIL n%<>-1 OR j%=max%
        LOOP UNTIL n%<>-1
        LINE x%(m%),y%(m%),smx%,smy%
        LINE x%(m%),y%(m%),x%(n%),y%(n%)
        DO
        LOOP UNTIL MOUSEK=0
        DO
          DO
            tst_mouse
          LOOP UNTIL mk%=1
          m%=-1
          i%=0
          DO
            IF x%(i%)>319 AND x%(i%)>mx%-6 AND
x%(i%)<mx%+6 AND y%(i%)>my%-6 AND y%(i%)<my%+6
              m%=i%
              dep2%=m%
            ENDIF
            INC i%
          LOOP UNTIL m%<>-1 OR i%=max%
        LOOP UNTIL m%<>-1
        DO
        LOOP UNTIL MOUSEK=0
        smx%=x%(m%)
        smy%=y%(m%)
        DO
          DO
            tst_mouse
            LINE x%(m%),y%(m%),smx%,smy%
            LINE x%(m%),y%(m%),mx%,my%
            smx%=mx%
            smy%=my%
          LOOP UNTIL mk%=1
          n%=-1
          j%=0
          DO
            IF x%(j%)>319 AND x%(j%)>mx%-6 AND
x%(j%)<mx%+6 AND y%(j%)>my%-6 AND y%(j%)<my%+6
              n%=j%
              arr2%=n%
            ENDIF
            INC j%
          LOOP UNTIL n%<>-1 OR j%=max%
        LOOP UNTIL n%<>-1
        LINE x%(m%),y%(m%),smx%,smy%
        LINE x%(m%),y%(m%),x%(n%),y%(n%)
        DO
        LOOP UNTIL MOUSEK=0
        i%=0
        c%=-1
        DO
          IF lig%(i%,0)=dep1% AND lig%(i%,1)=arr1%
AND lig%(i%,2)=dep2% AND lig%(i%,3)=arr2%
            lig%(i%,0)=-1
            c%=0
          ENDIF
          INC i%
        LOOP UNTIL c%=0 OR i%=max%
        i%=0
        c%=-1
        DO
          IF lig%(i%,0)=-1
            lig%(i%,0)=dep1%
            lig%(i%,1)=arr1%
            lig%(i%,2)=dep2%
            lig%(i%,3)=arr2%
            c%=0
          ENDIF
          INC i%
        LOOP UNTIL c%=0 OR i%=max%
      ENDIF
    ENDIF
  LOOP UNTIL mk%=2
  DEFMOUSE 0
  GRAPHMODE 1
  '
  scroll
  PRINT "Fin d'édition de ligne"
RETURN
PROCEDURE animate
  LOCAL i%,j%,x1%,y1%,x2%,y2%,adrbuf1%,adrbuf2%,
adrbuf3%,buffer1%,buffer2%,buffer3%,sauv_ecran%,
ecran%,logique%
  j%=-1
  '
  scroll
  scroll
  PRINT "Réalise l'interpolation entre l'image de
début et l'image de fin en une"
  scroll
  PRINT "en une vingtaine d'étapes."
  scroll
  scroll
  PRINT "Cliquez pour commencer..."
  '
```

suite en page 11

GRAPHISME Monochrome

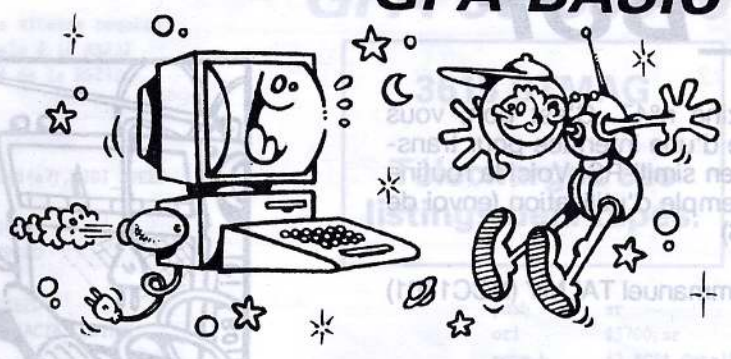
3615 STMAG 24H/24 100% ST

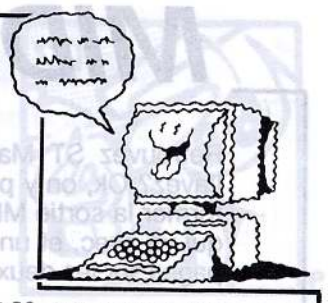
SPACE 19

GFA-BASIC

A bord de votre vaisseau spatial, dégommez tout obstacle fonçant sur vous. Entre chaque tableau, passez au magasin pour améliorer votre équipement (vous gagnez de l'argent pour chaque obstacle détruit).

DENEB (Zap Creation)






```
VOID XBIOS(7,0,0)*XBIOS(7,1,&H750)*XBIOS(7,2,&H640)
*XBIOS(7,3,&H530)*XBIOS(7,4,&H420)*XBIOS(7,5,&H310)
*XBIOS(7,15,&H777)*XBIOS(7,6,&H337)*XBIOS(7,7,&H447)
*XBIOS(7,8,&H557)
DEFTEXT 1,0,0,32
TEXT 80,90,"SPACE 19"
PRINT AT(7,20);"PAR DENEB(ZC) LE 31/5/91"
DO
  FOR i%=712 TO 789
    SETCOLOR 0,ADD(i%,m%)
  NEXT i%
  INC m%
  VSYNC
LOOP UNTIL INP(-2)
SETCOLOR 0,0
STICK 0
z%=XBIOS(2)
FOR i%=0 TO 16
  READ t%
  LPOKE (z%+i%*160),t%
  READ t%
  LPOKE (z%+4+i%*160),t%
  READ t%
  LPOKE (z%+8+i%*160),t%
  READ t%
  LPOKE (z%+12+i%*160),t%
  READ t%
  LPOKE (z%+16+i%*160),t%
NEXT i%
DATA 131078,65536,-2147418111,-2147352576
DATA -2147450880
DATA 655382,65536,1074135046,-1073676288,536928256
DATA 1703974,65536,537198597,-536215552,1073782784
DATA 3801158,65536,269025289,-268042240,268496896
DATA 3670080,7,786440,-267976704,-805253120
DATA 1081286787,196620,-2012839836
DATA -133496832,536879104,-1862831865,458760
DATA 1540342,-134152192,1610637312
DATA -1577623033,268894216,70353137,-134086656
DATA -2147450880
DATA -1544067833,268894216,35685600,-133169152,0
DATA -1476891889,269418496,18908896,-133169152,0
DATA -1074104570,65536,111150944,-133169152,0
DATA -1124434938,50397184,18874848,-32505856,0
DATA -1141212922,100728832,44040416,-15728640,0
DATA -1308987130,134283264,69206112,-40894464,0
DATA -1577426682,268500992,69206240,-66060288,0
DATA 569565454,268500992,69206240,-66060288,0
DATA -788020738,805371904,-53477184,-63963136,0
GET 0,0,27,16,s$
GET 28,0,35,7,b$
DEFTEXT 1,0,0,32
CLS
DIM iop|(32255)
il%=XBIOS(3)
i2%=(VARPTR(iop|(0))+255) AND &HFFFF00
n%=25
DIM vx%(n%),vy%(n%),ex%(n%),ey%(n%)
deb:
RESTORE dat
ly%=20
v%=4
arg%=0
en%=100
li%=15
FOR lev%=0 TO 10
  SETCOLOR 14,lev%*166
  FOR i%=0 TO 100+(lev%*25)
    PSET RANDOM(320),RANDOM(200),14
  NEXT i%
  SGET f$
  READ m%
  READ ve%
  READ vse%
  READ mtim%
mid:
  n%=m%
  FOR i%=0 TO n%
    vx%(i%)=RANDOM(ve%)-vse%
    IF vx%(i%)=0
      vx%(i%)=1
    ENDIF
    vy%(i%)=RANDOM(ve%)-vse%
    IF vy%(i%)=0
      vy%(i%)=1
    ENDIF
    ey%(i%)=0
    ex%(i%)=0
  NEXT i%
  CLR tim%
  x%=160
  y%=100
  oo%=0
  TEXT 100,90,"LEVEL "
  TEXT 130,125,lev%+1
  PRINT AT(14,20);"PRESS FIRE"
  REPEAT
  UNTIL STRIG(1)
  CLS
  DO
    SPUT f$
    @joy
    @int
    @en
    PRINT AT(1,1);"TIR:";en%;" SOU:";arg%;
" SCO:";sco%;" TIME:";mtim%-tim%;" VIE:";li%;" "
    PUT x%,y%,s$
    @sync
    INC tim%
    EXIT IF tim%=mtim% OR n%<0 OR oo%=1
  LOOP
  IF oo%=1 AND li%>0
    VOID XBIOS(5,L:il%,L:il%,-1)
    CLS
    en%=100
    GOTO mid
  ENDIF
  IF n%>=0 OR li%=0
    @go
    LOCATE 7,20
    INPUT "UNE AUTRE PARTIE ? (O/N)",ppp$
    EXIT IF ppp$<>"n" AND ppp$<>"N"
    EDIT
  ELSE
    VOID XBIOS(5,L:il%,L:il%,-1)
    DO
      EXIT IF mtim%-tim%<1
      INC tim%
      INC sco%
      PRINT AT(1,1);"TIR:";en%;" SOU:";arg%;
" SCO:";sco%;" TIME:";mtim%-tim%;" VIE:";li%;" "
    LOOP
    PAUSE 50
    @mag
  ENDIF
  PAUSE 20
NEXT lev%
CLS
GOTO deb
dat:
DATA 5,10,5,350
DATA 10,10,5,600
DATA 3,10,3,140
DATA 0,0,5,60
DATA 8,8,4,200
DATA 8,15,5,300
DATA 10,14,7,300
DATA 15,14,7,400
DATA 20,14,7,450
PROCEDURE sync
  SWAP il%,i2%
  VOID XBIOS(5,L:il%,L:i2%,-1)
RETURN
PROCEDURE joy
  IF STICK(1)=8 OR STICK(1)=9 OR STICK(1)=10 AND x%<290
    ADD x%,v%
  ELSE IF STICK(1)=4 OR STICK(1)=5 OR STICK(1)=6
AND x%>4
    SUB x%,v%
  ENDIF
  IF STICK(1)=2 OR STICK(1)=6 OR STICK(1)=10 AND
y%<184
    ADD y%,v%
  ELSE IF STICK(1)=1 OR STICK(1)=9 OR STICK(1)=5
AND y%>4
    SUB y%,v%
  ENDIF
  l!=FALSE
  IF STRIG(1) AND en%>0
    IF y%-ly%>0
      ALINE x%+14,y%-ly%,x%+14,y%,2,-1,1
    ELSE
      ALINE x%+14,0,x%+14,y%,2,-1,1
    ENDIF
    DEC en%
    l!=TRUE
  ENDIF
RETURN
PROCEDURE en
  FOR i%=0 TO n%
    IF ex%(i%)>330
      ex%(i%)=-8
    ENDIF
    IF ey%(i%)>208
      ey%(i%)=-8
    ENDIF
    IF ex%(i%)<-8
      ex%(i%)=330
    ENDIF
    IF ey%(i%)<-8
      ey%(i%)=208
    ENDIF
    ADD ex%(i%),vx%(i%)
    ADD ey%(i%),vy%(i%)
    PUT ex%(i%),ey%(i%),b$
  NEXT i%
RETURN
PROCEDURE int
  IF l!=TRUE
    FOR i%=0 TO n%
      tx%=x%+14
      ty%=y%-ly%
      tlx%=1
      tly%=ly%
      b!=RC_INTERSECT(ex%(i%),ey%(i%),8,8,tx%,ty%,
tlx%,tly%)
      IF b!
        SOUND 1,15,#10000
        PAUSE 1
        SOUND 1,0,0,0
        DELETE ex%(i%)
        DELETE ey%(i%)
        DELETE vx%(i%)
        DELETE vy%(i%)
        ADD arg%,10
        ADD sco%,10
        DEC n%
      ENDIF
    NEXT i%
  ENDIF
  FOR i%=0 TO n%
    tx%=x%
    ty%=y%
    tlx%=26
    tly%=16
    b!=RC_INTERSECT(ex%(i%),ey%(i%),8,8,tx%,ty%,
tlx%,tly%)
    IF b!
      DEC li%
      oo%=1
      PRINT CHR$(7)
    ENDIF
  NEXT i%
RETURN
PROCEDURE mag
  CLS
  PRINT AT(15,1);"MAGASIN"
  PRINT
  PRINT
  PRINT "1-enegie(+10)        10"
  PRINT "2-energie(+20)       15"
  PRINT "3-vitesse(+1)        10"
  PRINT "4-tire(+1)            3"
  PRINT
  PRINT "10-FIN"
r:
  LOCATE 1,15
  PRINT "ENERGIE(tir) : ";en%;" "
  PRINT "ARGENT : ";arg%;" "
  LOCATE 1,20
  INPUT "VOTRE CHOIX : ",ch%
  IF ch%=1 AND arg%>=10
    ADD en%,10
    SUB arg%,10
  ELSE IF ch%=2 AND arg%>=20
    ADD en%,20
    SUB arg%,20
  ELSE IF ch%=3 AND arg%>=10 AND v%<7
    ADD v%,1
    SUB arg%,10
  ELSE IF ch%=4 AND arg%>=3
    ADD ly%,1
    SUB arg%,3
  ENDIF
  IF ch%<>10
    GOTO r
  ENDIF
  CLS
RETURN
PROCEDURE go
  VOID XBIOS(5,L:il%,L:il%,-1)
  LOCATE 20,10
  TEXT 80,110,"GAME OVER"
RETURN
```

JEU Basse Résolution

INTERPOL

suite de la page 10

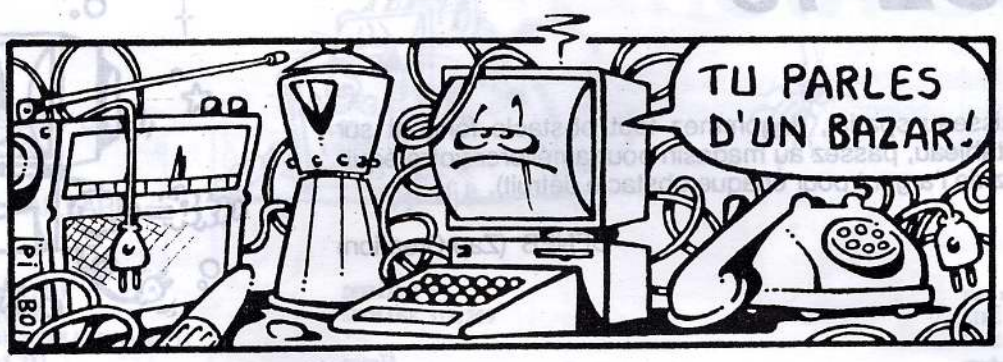
```
  FOR i%=0 TO max%
    IF lig%(i%,0)<>-1
      INC j%
      tab%(j%,0)=MUL(x%(lig%(i%,0)),2)
      tab%(j%,1)=MUL(SUB(y%(lig%(i%,0)),181),2)
      tab%(j%,2)=MUL(x%(lig%(i%,1)),2)
      tab%(j%,3)=MUL(SUB(y%(lig%(i%,1)),181),2)
      tab%(j%,4)=SUB(MUL(SUB(x%(lig%(i%,2)),320),
2),tab%(j%,0))
      tab%(j%,5)=SUB(MUL(SUB(y%(lig%(i%,2)),181),
2),tab%(j%,1))
      tab%(j%,6)=SUB(MUL(SUB(x%(lig%(i%,3)),320),
2),tab%(j%,2))
      tab%(j%,7)=SUB(MUL(SUB(y%(lig%(i%,3)),181),
2),tab%(j%,3))
    ENDIF
  NEXT i%
  '
  DO
  LOOP UNTIL MOUSEK=1
  DO
  LOOP UNTIL MOUSEK=0
  '
  sauv_ecran%=GEMDOS(72,L:32000)
  BMOVE XBIOS(2),sauv_ecran%,32000
  adrbuf1%=GEMDOS(72,L:32256)
  buffer1%=(adrbuf1% AND &HFFFF00)+256
  adrbuf2%=GEMDOS(72,L:32256)
  buffer2%=(adrbuf2% AND &HFFFF00)+256
  adrbuf3%=GEMDOS(72,L:32256)
  buffer3%=(adrbuf3% AND &HFFFF00)+256
  HIDEM
  ecran%=XBIOS(2)
  ~XBIOS(5,L:buffer1%,L:-1,W:-1)
  logique%=buffer1%
  CLOSEW 0
  '
  FOR k%=0 TO 20
    coef=1-COS(PI/2/20*k%)
    CLS
    FOR l%=0 TO j%
      x1%=tab%(l%,0)+coef*tab%(l%,4)
      y1%=tab%(l%,1)+coef*tab%(l%,5)
      x2%=tab%(l%,2)+coef*tab%(l%,6)
      y2%=tab%(l%,3)+coef*tab%(l%,7)
      LINE x1%,y1%,x2%,y2%
    NEXT l%
    IF logique%=buffer1%
      logique%=buffer2%
      ~XBIOS(5,L:buffer2%,L:buffer1%,W:-1)
    ELSE
      IF logique%=buffer2%
        logique%=buffer3%
        ~XBIOS(5,L:buffer3%,L:buffer2%,W:-1)
      ELSE
        logique%=buffer1%
        ~XBIOS(5,L:buffer1%,L:buffer3%,W:-1)
      ENDIF
    ENDIF
  NEXT k%
  OPENW 0
  '
  DO
    LOOP UNTIL MOUSEK=2
    '
    SHOWM
    ~XBIOS(5,L:ecran%,L:ecran%,W:-1)
    ~GEMDOS(73,L:adrbuf1%)
    ~GEMDOS(73,L:adrbuf2%)
    ~GEMDOS(73,L:adrbuf3%)
    BMOVE sauv_ecran%,XBIOS(2),32000
    ~GEMDOS(73,L:sauv_ecran%)
  RETURN
  PROCEDURE refresh
    LOCAL i%
    '
    LINE 0,179,639,179
    LINE 0,180,639,180
    BMOVE V:pimage$,ADD(XBIOS(2),16000),16000
    LINE 319,181,319,380
    LINE 320,181,320,380
    FOR i%=0 TO max%
      IF x%(i%)<>-1
        BOX x%(i%)-3,y%(i%)-3,x%(i%)+3,y%(i%)+3
      ENDIF
    NEXT i%
    FOR i%=0 TO max%
      IF lig%(i%,0)<>-1
        LINE x%(lig%(i%,0)),y%(lig%(i%,0)),x%(lig%
(i%,1)),y%(lig%(i%,1))
        LINE x%(lig%(i%,2)),y%(lig%(i%,2)),x%(lig%
(i%,3)),y%(lig%(i%,3))
      ENDIF
    NEXT i%
  RETURN
  PROCEDURE winselect
    DO
      scroll
      PRINT "Cliquez dans la fenêtre de votre
choix... ";
      '
      DO
      LOOP UNTIL MOUSEK=1
      DO
      LOOP UNTIL MOUSEK=0
      '
      IF MOUSEY>160
        IF MOUSEX<320
          dest%=V:pimage$
          c%=0
          '
          PRINT "Sélection de la fenetre de départ"
        ELSE IF MOUSEX>319
          dest%=V:pimage$+40
          c%=0
          '
          PRINT "Sélection de la fenêtre d'arrivée"
        ENDIF
      ENDIF
    LOOP UNTIL c%=0
  RETURN
  PROCEDURE tst_mouse
    MOUSE mx%,my%,mk%
    IF my%<181
      my%=181
      SETMOUSE mx%,my%+19
    ENDIF
  RETURN
```

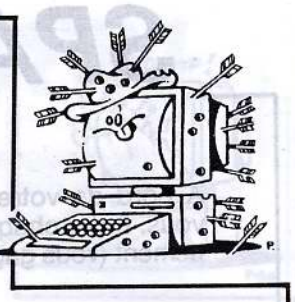
MIDI_BUF

GFA ou 68000

Retrouvez ST Magazine n°47. C'est bon, vous l'avez? Ok, on y parle d'une interface pour transformer la sortie MIDI en simili-RS. Voici la routine qui va avec, et un exemple d'utilisation (envoi de pages sur les deux RS)

Emmanuel TALMY (NCC1701)





```
REM ************************************************************
REM ** Exemple d'utilisation de la routine MIDI_BUF pour      **
REM ** gérer deux Minitels sur les interfaces RS232 et MIDI.  **
REM ** (C)Avril 1991                              par NCC1701  **
REM ************************************************************
'
$C+
RESERVE -30000              ! Libère 30Ko de mémoire
'
' Le INLINE ci-dessous contient la routine MIDI_BUF.INL
INLINE midi%,1104
' Définition des la taille des tampons utilisés pour l'exemple
in_size&=512                ! Tampons IN de 0,5 Ko
out_size&=3000              ! Tampons OUT de 3 Ko
' Chargement des fichiers VID de démonstration (10 maxi)
DIM vid_addr%(9),vid_length&(9),vid_name$(9)
REPEAT
  vid_count&=@vid_load      ! Charge les fichiers VID (10 maxi)
  IF vid_count&=0 THEN      ! SI aucun fichier VID,
    ALERT 3,"| Aucun fichier VID trouvé! |",1,"STOP|ENCORE",void&
  ENDIF
UNTIL vid_count&>0 OR void&=1
IF vid_count&>0 THEN        ! SI fichiers VID chargés,
  ' Installation et configuration des interfaces RS232 et MIDI
  rs_install(in_size&,out_size&)
  PRINT "Tapez sur le Minitel la numéro de page souhaité...";
  TITLEW #1," PAGES ENVOYEES (clic pour arrêter) "
  OPENW #1,0,WORK_OUT(1)\2,WORK_OUT(0),WORK_OUT(1)\2,&X1
  ' Boucle principale :
  ' On teste l'arrivée d'un caractère sur les deux interfaces
  ' RS232 et MIDI. Si c'est un chiffre correspondant à une des
  ' pages VID chargées, on envoie cette page sur l'interface.
  REPEAT                    ! Boucle
    input_test(1)           ! Teste la RS232 (port 1)
    input_test(3)           ! Teste la MIDI (port 3)
  UNTIL MOUSEK              ! Arrêt si clic souris
  CLOSEW #1
  CLOSEW 0
  CLS
  esc$=CHR$(27)
  ' Libère la mémoire des images VID
  vid_unload
  ' Restaure l'état standard des interfaces RS232 et MIDI
  rs_desinstall
ENDIF
RESERVE
END
'
FUNCTION vid_load           ! Recherche et charge les fichiers VID
  LOCAL dta_addr%,e&,vid_index&
  CLS
  PRINT esc$;"p Recherche et chargement des fichiers VID ";esc$;"q"
  PRINT
  vid_index&=0
  dta_addr%=FGETDTA()
  e&=FSFIRST("*.VID",0)
  DO WHILE e&=0
    vid_name$(vid_index&)=CHAR{dta_addr%+30}
    OPEN "I",#1,vid_name$(vid_index&)
    vid_length&(vid_index&)=LOF(#1)
    IF MALLOC(-1)>LOF(#1) THEN
      PRINT "Fichier n°";vid_index&;TAB(15);vid_name$
 (vid_index&);TAB(30);vid_length&(vid_index&);" octets";
      vid_addr%(vid_index&)=MALLOC(LOF(#1))
      BGET #1,vid_addr%(vid_index&),LOF(#1)
      INC vid_index&
      PRINT "...OK !"
    ENDIF
    CLOSE #1
    e&=FSNEXT()
  LOOP UNTIL vid_index&=10
  RETURN vid_index&
ENDFUNC
PROCEDURE vid_unload        ! Supprime les fichiers VID en mémoire
  LOCAL vid_index&
  vid_index&=vid_count&
  WHILE vid_index&>0
    DEC vid_index&
    ~MFREE(vid_addr%(vid_index&))
  WEND
RETURN
' ** Les deux Minitels sont supposés être à 1200 bauds au départ
PROCEDURE rs_install(in_size&,out_size&)! Installation globale
  LOCAL conf&,in_buffer%,out_buffer%
  ' ** Initialisation RS232
  @save_rs232               ! Mémorise les tampons standards RS232
  in_buffer%=MALLOC(in_size&)   ! Alloue le tampon IN RS232
  {XBIOS(14,0)}=in_buffer%      ! Adresse et taille tampon IN
  INT{XBIOS(14,0)+4}=in_size&   ! dans le IOREC RS232
  out_buffer%=MALLOC(out_size&) ! Alloue le tampon OUT RS232
  {XBIOS(14,0)+14}=out_buffer%  ! Adresse et taille tampon OUT
  INT{XBIOS(14,0)+18}=out_size& ! dans le IOREC RS232
  ' ** Initialisation MIDI
  in_buffer%=MALLOC(in_size&)   ! Alloue le tampon IN MIDI
  out_buffer%=MALLOC(out_size&) ! Alloue le tampon OUT MIDI
  ' ** Choix de la configuration des vitesses
  conf&=FORM_ALERT(0,"[0][|    Configuration RS232-MIDI    |]
 [1200-1200|4800-1200|4800-4800]")
  rs232_speed(1200)             ! RS232 à 1200 bauds par défaut
  SELECT conf&
  CASE 1                        ! 1200/1200
    ' ** MIDI à 1200 bauds (= RS232)
    ~C:midi%(1,0,2,L:in_buffer%,in_size&,L:out_buffer%,out_size&)
  CASE 2                        ! 4800/1200
    set_speed(1,4800)           ! Minitel RS232 à 4800 bauds
    rs232_speed(4800)           ! RS232 à 4800 bauds
    ' ** MIDI à 4800/4=1200 bauds
    ~C:midi%(2,0,2,L:in_buffer%,in_size&,L:out_buffer%,out_size&)
  CASE 3                        ! 4800/4800
    ' ** MIDI à 1200 bauds (=RS232)
    ~C:midi%(1,0,2,L:in_buffer%,in_size&,L:out_buffer%,out_size&)
    set_speed(1,4800)           ! Minitel RS232 à 4800 bauds
    set_speed(3,4800)           ! Minitel MIDI à 4800 bauds
    rs232_speed(4800)           ! RS232 et MIDI à 4800 bauds
  ENDSELECT
  set_screen(1,FALSE)
  @cls(1)                       ! Efface écran Minitel RS232
  set_screen(3,FALSE)
  @cls(3)                       ! Efface écran Minitel MIDI
RETURN
'
PROCEDURE rs_desinstall     ! Restaure les ports standards
  desinstall(1)             ! Libère les tampons RS232
  @restore_rs232            ! Restaure les tampons standards RS232
  desinstall(3)             ! Libère les tampons MIDI
  rs232_speed(1200)         ! RS232 à 1200 bauds
  ~C:midi%(-1)              ! Restaure le standard MIDI
RETURN
PROCEDURE desinstall(port&)     ! Restaure l'état standard du
port
  set_screen(port&,TRUE)        ! Reconnecte l'écran du Minitel
  set_speed(port&,1200)         ! Remet le Minitel en 1200 bauds
  REPEAT                        ! Attend le vidage du tampon OUT
  UNTIL @out_empty(port&)
  ~MFREE({XBIOS(14,PRED(port&))}) ! Libère le tampon IN du port
  ~MFREE({XBIOS(14,PRED(port&))+14}) ! Libère le tampon OUT
RETURN
'
PROCEDURE input_test(port&)     ! Teste entrées depuis un Minitel
  LOCAL vid_index&
  ' SI char reçu sur le port et que le tampon de sortie est vide,
  IF INP?(port&) AND @out_empty(port&) THEN
    vid_index&=INP(port&)-48    ! Lecture du char reçu
    IF vid_index&>=0 AND vid_index&<vid_count& THEN
      ' SI c'est un numéro OK,
      vid_output(port&,vid_index&)
    ELSE                        ! SINON,
      OUT port&,7               ! beep vers le Minitel
    ENDIF
  ENDIF
RETURN
PROCEDURE vid_output(port&,vid_index&) ! Sortie page vers un mntl
  LOCAL start_addr%,end_addr%,addr%
  PRINT "Emission de ";vid_name$(vid_index&);" sur la sortie ";
  IF port&=1 THEN
    PRINT "RS232...";
  ELSE
    PRINT "MIDI....";
  ENDIF
  start_addr%=vid_addr%(vid_index&)
  end_addr%=start_addr%+vid_length&(vid_index&)-1
  FOR addr%=start_addr% TO end_addr%
    OUT port&,BYTE{addr%}
  NEXT addr%
  PRINT "terminée !"
RETURN
PROCEDURE save_rs232        ! Sauvegarde adresses RS232 standards
  old_in%={XBIOS(14,0)}
  old_in&=INT{XBIOS(14,0)+4}
  old_out%={XBIOS(14,0)+14}
  old_out&=INT{XBIOS(14,0)+18}
RETURN
PROCEDURE restore_rs232     ! Restaure adresses RS232 standards
  {XBIOS(14,0)}=old_in%
  INT{XBIOS(14,0)+4}=old_in&
  {XBIOS(14,0)+14}=old_out%
  INT{XBIOS(14,0)+18}=old_out&
RETURN
FUNCTION out_empty(port&)   ! Teste si un tampon OUT est vide
  RETURN
INT{XBIOS(14,PRED(port&))+20}=INT{XBIOS(14,PRED(port&))+22}
ENDFUNC
'
PROCEDURE rs232_speed(speed&)     ! Modifie la vitesse RS232
  SELECT speed&
  CASE 75
    speed&=14
  CASE 300
    speed&=9
  CASE 4800
    speed&=2
  DEFAULT
    speed&=7
  ENDSELECT
  ~XBIOS(15,speed&,0,174,-1,-1,-1) ! Configuration pour le mntl
  PAUSE 5
  WHILE INP?(1)                    ! Vide tampon RS232 IN
    ~INP(1)
  WEND
RETURN
PROCEDURE set_speed(port&,speed&)  ! Modifie la vitesse du mntl
SELECT speed&
  CASE 75
    speed&=&X1001001
  CASE 300
    speed&=&X1010010
  CASE 4800
    speed&=&X1110110
  DEFAULT
    speed&=&X1100100
  ENDSELECT
  OUT port&,&H1B,&H3A,&H6B,speed&   ! Envoi de la commande
  PAUSE 5
RETURN
' Modifie l'aiguillage Modem-Ecran
PROCEDURE set_screen(port&,status!)
  OUT port&,&H1B,&H3B,&H60-status!,&H58,&H52
  void_inp(port&,5)                 ! Attend l'acquittement
RETURN
PROCEDURE cls(port&)                ! Efface écran Minitel
  OUT port&,12
RETURN
PROCEDURE void_inp(port&,count&)    ! Lecture des acquittements
  IF count&<=0 THEN                 ! SI count=-1
    WHILE INP?(port&)               ! vide le tampon IN
    WEND
  ELSE                              ! SI count>0,
    WHILE count&>0                  ! lit le nombre d'octets requis
      ~INP(port&)
      DEC count&
    WEND
  ENDIF
RETURN
'
' ******************************
' * Programme créant le fichier *
' *  MIDIBUF.INL, nécessaire à  *
' *  l'INLINE du programme GFA  *
' * (pour ceusses qui n'ont pas *
' *   d'assembleur avec eux...  *
' ******************************
'
yepa_ligne=0
OPEN "o",#1,"MIDIBUF.INL"
DO
  INC yepa_ligne
  yepa_check=0
  FOR yepa_yepa=1 TO 16
    READ yepa_donnee$
    yepa_donnee=VAL("&h"+yepa_donnee$)
    EXIT IF yepa_donnee$="-1"
    ADD yepa_check,yepa_donnee
    OUT #1,yepa_donnee
  NEXT yepa_yepa
  EXIT IF yepa_donnee$="-1"
  READ yepa_ctrl$
  IF VAL("&h"+yepa_ctrl$)<>yepa_check
    PRINT "Erreur de DATA en ligne ";yepa_ligne;"."
    PRINT "Vous n'êtes pas concentré..."
    END
  ENDIF
  PRINT "Ligne ";yepa_ligne;" exploitÇe...";CHR$(13);
LOOP
CLOSE #1
DATA 60,1A,0,0,4,30,0,0,0,0,0,0,0,0,0,0,AE
DATA 0,0,0,0,0,0,0,0,0,0,0,0,70,FF,4D,FA,2B6
DATA 4,0,4A,6F,0,4,6B,0,0,F2,4A,2E,0,29,66,0,325
DATA 0,E8,3F,3C,0,22,4E,4E,54,4F,20,40,2D,68,0,1C,3D5
DATA 0,0,41,E8,0,1C,2D,48,0,4,1E,3C,0,80,30,2F,2F7
DATA 0,4,6E,2,70,1,C,40,0,2,6F,2,70,1,8E,0,2A3
DATA 30,2F,0,8,6A,2,70,2,C,40,0,7,6F,2,70,2,27B
DATA E5,40,8E,0,30,2F,0,6,6A,2,42,0,C,0,0,2,2D4
DATA 6F,2,42,0,1D,40,0,28,C,0,0,2,66,4,8,C7,27F
DATA 0,6,1D,47,0,2A,2D,6F,0,A,0,8,3D,6F,0,E,1FC
DATA 0,C,42,6E,0,E,42,6E,0,10,3D,7C,0,40,0,12,295
DATA 3D,6F,0,E,0,14,4,6E,0,40,0,14,2D,6F,0,10,240
DATA 0,16,3D,6F,0,14,0,1A,42,6E,0,1C,42,6E,0,1E,28A
DATA 3D,7C,0,40,0,20,3D,6F,0,14,0,22,4,6E,0,40,2AD
DATA 0,22,48,7A,0,AC,48,79,0,5,0,2D,4E,4D,50,4F,3BD
DATA 2D,40,FD,C6,48,7A,0,F4,48,79,0,5,0,2E,4E,4D,575
DATA 50,4F,2D,40,FE,A,48,7A,0,48,3F,3C,0,26,4E,4E,45B
DATA 5C,4F,50,EE,0,29,70,0,4E,75,4A,2E,0,29,67,2E,47B
DATA 48,7A,0,4E,3F,3C,0,26,4E,4E,5C,4F,2F,3A,0,B8,419
DATA 48,79,0,5,0,2D,4E,4D,50,4F,2F,3A,0,EE,48,79,445
DATA 0,5,0,2E,4E,4D,50,4F,51,EE,0,29,70,0,4E,75,408
DATA 40,E7,0,7C,7,0,11,FC,0,3,FC,4,20,6E,0,4,44C
DATA 43,FA,1,C8,20,89,11,EE,0,2A,FC,4,46,DF,4E,75,6C0
DATA 40,E7,0,7C,7,0,11,FC,0,3,FC,4,20,6E,0,4,44C
DATA 22,6E,0,0,20,89,11,FC,0,95,FC,4,46,DF,4E,75,5C3
DATA 41,EF,0,6,8,17,0,5,66,2,4E,68,C,90,0,1,315
DATA 0,3,66,6,61,0,0,88,4E,73,C,90,0,2,0,3,2BA
DATA 66,E,61,0,0,7A,4A,80,67,F8,61,0,0,9C,4E,73,536
DATA C,90,0,3,0,3,66,E,30,28,0,4,61,0,0,F4,2C7
DATA 4A,80,6B,F4,4E,73,C,90,0,8,0,3,66,6,61,0,45E
DATA 0,5E,4E,73,4E,F9,0,1,9,26,41,EF,0,6,8,17,3EB
DATA 0,5,66,2,4E,68,C,50,0,C,66,1C,32,28,0,2,269
DATA 20,68,0,4,10,18,3F,0,61,0,0,B8,30,1F,4A,80,325
DATA 6B,F4,51,C9,FF,F0,4E,73,C,90,0,E,0,2,66,8,643
DATA 41,FA,2,6,20,8,4E,73,4E,F9,0,FC,7,F2,30,3A,5D2
DATA 1,FE,B0,7A,1,FC,56,C0,48,80,48,C0,4E,75,30,3A,739
DATA 1,FE,52,40,B0,7A,1,F4,66,2,70,0,B0,7A,1,EE,6A1
DATA 56,C0,48,80,48,C0,4E,75,40,E7,0,7C,7,0,48,E7,682
DATA 40,82,4D,FA,1,BC,32,2E,0,E,B2,6E,0,10,67,1A,4E5
DATA 52,41,B2,6E,0,C,65,2,72,0,70,0,20,6E,0,8,39E
DATA 10,30,10,0,3D,41,0,E,60,2,70,FF,4A,2E,0,26,34B
DATA 67,28,32,2E,0,10,92,6E,0,E,64,4,D2,6E,0,C,3C1
DATA B2,6E,0,12,66,14,2F,0,10,3C,0,11,61,0,0,14,2AD
DATA 4A,80,6B,F4,20,1F,51,EE,0,26,4C,DF,41,2,46,DF,660
DATA 4E,75,40,E7,0,7C,7,0,48,E7,40,82,4D,FA,1,52,5F8
DATA 4A,2E,0,27,66,2A,32,2E,0,1E,B2,6E,0,1C,66,20,36F
DATA 8,38,0,1,FC,4,67,18,11,C0,FC,6,12,2E,0,2A,3FD
```

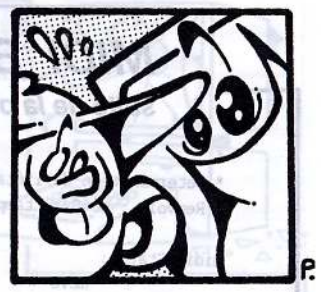
DISCUTER AVEC NCC
1701: 3615 STMAG

TELEMATIQUE
Toutes Résolutions

suite en page 13

3615 STMAG

Téléchargez ces listings déjà tapés.



MIDI_BUF

suite de la page 12

```
DATA 8,81,0,6,8,C1,0,5,11,C1,FC,4,70,0,60,22,421
DATA 52,41,B2,6E,0,1A,65,2,72,0,B2,6E,0,1C,67,10,459
DATA 20,6E,0,16,11,80,10,0,3D,41,0,1E,70,0,60,2,2B3
DATA 70,FF,4C,DF,41,2,46,DF,4E,75,48,E7,C0,82,4D,FA,87D
DATA 0,F0,10,38,FC,4,8,0,0,7,67,32,8,0,0,0,2E8
DATA 67,C,1D,40,0,24,3F,0,61,0,0,62,30,1F,8,0,24D
DATA 0,1,67,C,1D,40,0,25,3F,0,61,0,0,1E,30,1F,203
DATA 8,0,0,5,67,8,3F,0,61,0,0,A,30,1F,4C,DF,2A0
DATA 41,3,4E,75,10,38,FC,6,4E,75,4A,2E,0,27,66,24,43D
DATA 32,2E,0,1C,B2,6E,0,1E,67,1A,52,41,B2,6E,0,1A,408
DATA 65,2,42,41,20,6E,0,16,11,F0,10,0,FC,6,3D,41,41F
DATA 0,1C,60,6,11,EE,0,2A,FC,4,4E,75,10,38,FC,6,4B8
DATA C,2E,0,1,0,28,66,18,C,0,0,13,66,6,50,EE,2AA
DATA 0,27,60,5A,C,0,0,11,66,6,51,EE,0,27,60,4E,37E
DATA 32,2E,0,10,52,41,B2,6E,0,C,65,2,42,41,B2,6E,439
DATA 0,E,67,3A,20,6E,0,8,11,80,10,0,3D,41,0,10,274
DATA C,2E,0,1,0,28,66,26,4A,2E,0,26,66,20,92,6E,313
DATA 0,E,64,4,D2,6E,0,C,B2,6E,0,14,66,10,10,3C,3B8
DATA 0,13,61,0,FE,AE,4A,80,6B,F4,50,EE,0,26,4E,75,670
DATA 0,FC,29,FC,0,0,E,2E,0,7,14,94,2,0,0,A,318
DATA 0,13,0,40,1,C0,0,7,16,94,B,B8,5,2D,5,2D,2EC
DATA 0,40,B,78,83,83,0,0,0,0,89,90,0,0,0,0,2E2
DATA -1

*****************************************************
* Gestion de tampons E/S pour l'interface MIDI.     *
* Complément à l'article "DOUBLE-RS" de ST-MAG 47.  *
* (C) Mars 1991                          NCC1701    *
*****************************************************

* Adresses-systèmes utilisées
ACIA_Cntrl     =$FFFFFC04  Registre de controle MIDI
ACIA_Status    =$FFFFFC04  Registre d'état MIDI
ACIA_Data      =$FFFFFC06  Registre de données MIDI

* Fonctions BIOS/XBIOS utilisées
BConStat=1     Etat d'entrée d'un périphérique
ConIn=2        Lecture sur un périphérique
BConOut=3      Sortie sur un périphérique
SetExec=5      Revectorisation des exceptions
BCoStat=8      Etat de sortie d'un périphérique
IORec=14       Adresse des IOREC RS232/Clavier/MIDI
MidiWS=12      Envoi d'un chaine au MIDI
KbdVBase=34    Vecteur des interruptions Clavier/Midi
SupExec=38     Appel d'une routine en mode superviseur

* Codes de commandes du protocole XON-XOFF
XON=$11
XOFF=$13

* Des Macros pour la sauvegarde des registres
push           MACRO
               move.\0 \1,-(a7)
               ENDM
pop            MACRO
               move.\0 (a7)+,\1
               ENDM
pushm          MACRO
               movem.\0 \1,-(a7)
               ENDM
popm           MACRO
               movem.\0 (a7)+,\1
               ENDM

* Liste des paramètres attendus sur la pile
               RSRESET
               rs.l    1
_midi_speed    rs.w    1          Code vitesse de transfert
_midi_handshake rs.w   1          Flag XON-XOFF
_midi_format   rs.w    1          Format des données
_midi_inbuffer rs.l    1          Adresse du tampon d'entrée
_midi_insize   rs.w    1          Taille du tampon d'entrée
_midi_outbuffer rs.l   1          Adresse du tampon de sortie
_midi_outsize  rs.w    1          Taille du tampon de sortie
*
RETURN_CODE    EQUR    D0
VARBASE        EQUR    A6
*
               moveq   #-1,RETURN_CODE     Code erreur par défaut
               lea     VarBase(pc),VARBASE Base des variables
               tst     _midi_speed(a7)
               bmi     MidiDesinstall

* Mise en place de la gestion de l'ACIA MIDI
MidiInstall:
               tst.b   MidiFlag(VARBASE)   Si déjà installé,
               bne     .error              retour avec erreur

* Sauvegarde du vecteur d'interruption standard
               push    #KbdVBase
               trap    #14
               addq.w  #2,a7
               move.l  d0,a0
               move.l  28(a0),StndMidiVec(VARBASE)
               lea     28(a0),a0
               move.l  a0,AddrMidiVec(VARBASE)

* Initialise octet de contrôle ACIA,
* IRQ du récepteur activée, IRQ du transmetteur inhibée
*
ACIA_CONTROL   EQUR    D7
*
               move.b  #$80,ACIA_CONTROL

* Bits 01 de controle ACIA selon la vitesse requise
* Valeur 1, horloge/16, vitesse égale à la RS232
* Valeur 2, horloge/64, vitesse 1/4 de la RS232
*
MIDI_SPEED     EQUR    D0
*
               move    _midi_speed(a7),MIDI_SPEED
               bgt.s   .speed1
               moveq   #1,MIDI_SPEED
.speed1        cmpi    #2,MIDI_SPEED
               ble.s   .speed2
               moveq   #1,MIDI_SPEED
.speed2        or.b    MIDI_SPEED,ACIA_CONTROL

* Bits 234 du controle ACIA selon le format requis
* Valeur 2 par défaut, pour le minitel
*
MIDI_FORMAT    EQUR    D0
*
               move    _midi_format(a7),MIDI_FORMAT
               bpl.s   .form1
               moveq   #2,MIDI_FORMAT
.form1         cmpi    #7,MIDI_FORMAT
               ble.s   .form2
               moveq   #2,MIDI_FORMAT
.form2         asl     #2,MIDI_FORMAT
               or.b    MIDI_FORMAT,ACIA_CONTROL

* Marque drapeau de protocole (0=rien/1=XON-XOFF/2=RTS_CTS)
*
MIDI_HAND      EQUR    D0
*
               move    _midi_handshake(a7),MIDI_HAND
               bpl.s   .hand1
               clr.b   MIDI_HAND
.hand1         cmpi.b  #2,MIDI_HAND
               ble.s   .hand2
               clr.b   MIDI_HAND
.hand2         move.b  MIDI_HAND,XMode(VARBASE)
               cmpi.b  #2,MIDI_HAND        Si RTS/CTS,
               bne.s   .hand3              la broche RTS sera
               bset    #6,ACIA_CONTROL     haute hors-émission
.hand3:
               move.b  ACIA_CONTROL,ACIACntrl(VARBASE)

* Initialisation du IOREC d'entrée
               move.l  _midi_inbuffer(a7),InBuffer(VARBASE)
               move    _midi_insize(a7),InSize(VARBASE)
               clr     InHead(VARBASE)
               clr     InTail(VARBASE)
               move    #64,InLow(VARBASE)
               move    _midi_insize(a7),InHigh(VARBASE)
               subi    #64,InHigh(VARBASE)

* Initialisation du IOREC de sortie
               move.l  _midi_outbuffer(a7),OutBuffer(VARBASE)
               move    _midi_outsize(a7),OutSize(VARBASE)
               clr     OutHead(VARBASE)
               clr     OutTail(VARBASE)
               move    #64,OutLow(VARBASE)
               move    _midi_outsize(a7),OutHigh(VARBASE)
               subi    #64,OutHigh(VARBASE)

* Revectorise le BIOS et le XBIOS
               pea     BIOS(pc)
               pea     SetExec<<16+45
               trap    #13
               addq.w  #8,a7
               move.l  d0,StndBIOS+2-VarBase(VARBASE)
               pea     XBIOS(pc)
               pea     SetExec<<16+46
               trap    #13
               addq.w  #8,a7
               move.l  d0,StndXBIOS+2-VarBase(VARBASE)

* Initialise l'ACIA et l'interruption MIDI
               pea     SetACIA(pc)
               push    #SupExec
               trap    #14
               addq.w  #6,a7

* Marque routines installées, pas d'erreur
               st      MidiFlag(VARBASE)
               moveq   #0,RETURN_CODE
.error         rts

* Remise en place de la gestion standard du MIDI
MidiDesinstall:
               tst.b   MidiFlag(VARBASE)   Déjà désinstallé ?
               beq.s   .error              retour avec erreur

* Réinstalle l'ACIA et l'interruption MIDI standards
               pea     ResetACIA(pc)
               push    #SupExec
               trap    #14
               addq.w  #6,a7

* Réinstalle les vecteurs BIOS et XBIOS standards
               push.l  StndBIOS+2(pc)
               pea     SetExec<<16+45
               trap    #13
               addq.w  #8,a7
               push.l  StndXBIOS+2(pc)
               pea     SetExec<<16+46
               trap    #13
               addq.w  #8,a7

* Marque routines désinstallées, pas d'erreur
               sf      MidiFlag(VARBASE)
               moveq   #0,RETURN_CODE
.error         rts

* Mise en place des nouveaux paramètres ACIA
SetACIA:
               push    sr                  IPL=7
               ori     #$700,sr
               move.b  #3,ACIA_Cntrl\w     Reset ACIA
               move.l  AddrMidiVec(VARBASE),a0
               lea     MidiVec(pc),a1      Interruption MIDI
               move.l  a1,(a0)
               move.b  ACIACntrl(VARBASE),ACIA_Cntrl\w  Init ACIA
               pop     sr
               rts
* Remise en place des paramètres standard MIDI
ResetACIA:
               push    sr                  IPL=7
               ori     #$700,sr
               move.b  #3,ACIA_Cntrl\w     Reset ACIA
               move.l  AddrMidiVec(VARBASE),a0
               move.l  StndMidiVec(VARBASE),a1
               move.l  a1,(a0)             Interruption MIDI
               move.b  #$95,ACIA_Cntrl\w   ACIA MIDI
               pop     sr
               rts

* Nouveau vecteur BIOS gérant les fonctions MIDI
*
PARM_LIST      EQUR    A0
*
BIOS           lea     6(a7),PARM_LIST     Adresse paramètres
               btst    #5,(a7)             en mode superviseur
               bne.s   bios1
               move.l  usp,PARM_LIST       ou utilisateur

* Etat d'entrée du MIDI ?
bios1          cmpi.l  #BConStat<<16+3,(PARM_LIST)
               bne.s   bios2
               bsr     MidiInStatus        Lecture de l'état
               rte

* Attente d'un caractère sur le MIDI ?
bios2          cmpi.l  #ConIn<<16+3,(PARM_LIST)
               bne.s   bios3
.wait          bsr     MidiInStatus        Attend la présence
               tst.l   d0                  d'un caractère
               beq.s   .wait
               bsr     MidiGet             Lecture du caractère
               rte

* Envoi d'un caractère au MIDI ?
bios3          cmpi.l  #BConOut<<16+3,(PARM_LIST)
               bne.s   bios8
.send          move    4(PARM_LIST),d0
Tentative d'émission
               bsr     MidiPut
               tst.l   d0                  jusqu'à ce que
               bmi.s   .send               ça passe
               rte

* Etat de sortie du MIDI ?
bios8          cmpi.l  #BCoStat<<16+3,(PARM_LIST)
               bne.s   StndBIOS
               bsr     MidiOutStatus       Lecture de l'état
               rte

* Saut au BIOS standard pour les autres fonctions
StndBIOS:
               jmp     $87654321

* Nouveau vecteur XBIOS gérant les fonctions MIDI
XBIOS          lea     6(a7),PARM_LIST     Liste des paramètres
               btst    #5,(a7)             en mode superviseur
               bne.s   xbios12
               move.l  usp,PARM_LIST       ou utilisateur

* Envoi d'une chaine de caractères au MIDI?
*
CHAR_DECOUNT   EQUR    D1
STRING_ADDR    EQUR    A0
*
xbios12        cmpi    #MidiWS,(PARM_LIST)
               bne.s   xbios14
               move    2(PARM_LIST),CHAR_DECOUNT
décompteur
               move.l  4(PARM_LIST),STRING_ADDR adresse
.char          move.b  (STRING_ADDR)+,d0   caractère courant
.send          push    d0
               bsr     MidiPut             envoyé au MIDI
               pop     d0
               tst.l   d0
               bmi.s   .send
               dbra    CHAR_DECOUNT,.char  boucle
               rte

* Demande de l'adresse de l'IOREC du MIDI?
xbios14        cmpi.l  #IORec<<16+2,(PARM_LIST)
               bne.s   StndXBIOS
               lea     VarBase+MidiIORec(pc),a0 Adresse
               move.l  a0,d0               de l'IOREC
interne
               rte

* Saut au XBIOS standard pour les autres fonctions
StndXBIOS
               jmp     $87654321
```

suite en page 14

MIDI_BUF

suite de la page 13

```
*
* Détermination de l'état d'entrée du MIDI
* Renvoie : d0=0 si tampon vide, -1 sinon
*
MidiInStatus:
        move    VarBase+InHead(pc),d0
        cmp     VarBase+InTail(pc),d0
        sne     d0
        ext.w   d0
        ext.l   d0
        rts
*
* Détermination de l'état de sortie du MIDI
* Renvoie : d0=0 si tampon plein, -1 sinon
*
MidiOutStatus:
        move    VarBase+OutTail(pc),d0
        addq    #1,d0
        cmp     VarBase+OutSize(pc),d0
        bne.s   .star
        moveq   #0,d0
.star   cmp     VarBase+OutHead(pc),d0
        sne     d0
        ext.w   d0
        ext.l   d0
        rts
*
* Lecture d'un octet dans le tampon d'entrée MIDI
* Renvoie : d0=caractère lu ou -1 si tampon d'entrée vide
*
MidiGet     push    sr                      IPL=7
            or      #$700,sr
            pushm.l d1/a0/VARBASE           Sauvegarde registres
            lea     VarBase(pc),VARBASE
            move    InHead(VARBASE),d1      SI Head(IN)<>Tail(IN),
            cmp     InTail(VARBASE),d1
            beq.s   .get02
            addq    #1,d1                   Wrap(Head(IN))
            cmp     InSize(VARBASE),d1
            bcs.s   .get01
            moveq   #0,d1
.get01      moveq   #0,d0                   d0>0, char lu
            move.l  InBuffer(VARBASE),a0
            move.b  0(a0,d1.w),d0
            move    d1,InHead(VARBASE)      Update Head(IN)
            bra.s   .get10
.get02      moveq   #-1,d0                  SINON, d0<0, pas de char
.get10      tst.b   RxOff(VARBASE)          SI réception inhibée,
            beq.s   .get12
            move    InTail(VARBASE),d1      ET SI Tail-Head=Low
water,
            sub     InHead(VARBASE),d1
            bcc.s   .get11
            add     InSize(VARBASE),d1
.get11      cmp     InLow(VARBASE),d1
            bne.s   .get12
            push.l  d0                      Envoie code XON
.loop       move.b  #XON,d0                 à l'émetteur
            bsr     MidiPut
            tst.l   d0
            bmi.s   .loop
            pop.l   d0
            sf      RxOff(VARBASE)          Réception activée
.get12      popm.l  d1/a0/VARBASE           Restaure registres
            pop     sr                      et restaure IPL
            rts
*
* Ecriture d'un octet dans le tampon de sortie MIDI
* Entrée  : d0=caractère à émettre
* Renvoie : d0=0 si car. transmis,-1 si tampon de sortie plein
*
MidiPut     push    sr                      IPL=7
            ori     #$700,sr
            pushm.l d1/a0/VARBASE           Sauvegarde registres
            lea     VarBase(pc),VARBASE     A6=base des variables
            tst.b   TxOff(VARBASE)          SI transmission autorisée,
            bne.s   .put01
            move    OutTail(VARBASE),d1     ET SI tampon de
            cmp     OutHead(VARBASE),d1     sortie vide,
            bne.s   .put01
            btst    #1,ACIA_Status\w        ET SI transmetteur ACIA
libre,
            beq.s   .put01
            move.b  d0,ACIA_Data\w          écriture directe dans l'ACIA
            move.b  ACIACntrl(VARBASE),d1   Octet de cntrl ACIA
            bclr    #6,d1                   Force ligne RTS basse
            bset    #5,d1                   Demande une IRQ sur TX Empty
            move.b  d1,ACIA_Cntrl\w
            moveq   #0,d0                   renvoie d0=0, char transmis
            bra.s   .put99
.put01      addq    #1,d1                   wrap(Tail(OUT))
            cmp     OutSize(VARBASE),d1
            bcs.s   .put10
            moveq   #0,d1
.put10      cmp     OutHead(VARBASE),d1     SI tail<>head,
            beq.s   .put20                  char dans le
            move.l  OutBuffer(VARBASE),a0   tampon de sortie
            move.b  d0,0(a0,d1.w)
            move    d1,OutTail(VARBASE)     update tail
            moveq   #0,d0                   renvoie d0=0,
            bra.s   .put99                  char transmis
.put20      moveq   #-1,d0                  SINON, d0<0, char non-transmis
.put99      popm.l  d1/a0/VARBASE           Restaure les registres
            pop     sr                      et les interruptions
            rts
*
* Routine d'interruption traitant les IRQ de l'ACIA MIDI
*
ACIA_STATUS EQUR    D0
*
MidiVec     pushm.l d0-d1/a0/VARBASE        Sauvegarde registres
            lea     VarBase(pc),VARBASE
            move.b  ACIA_Status\w,ACIA_STATUS   Lecture du status ACIA
            btst    #7,ACIA_STATUS          SI b7 du status ACIA=0,
            beq.s   MidiVecExit             ce n'est pas une IT du MIDI
            btst    #0,ACIA_STATUS          Interruption du receveur?
            beq.s   .Vec01
            move.b  ACIA_STATUS,RxStatus(VARBASE)   Mémorise état receveur
            push    ACIA_STATUS             Sauver l'état
            bsr     RxFull                  -> Lire le caractère reçu
            pop     ACIA_STATUS             Restaure l'état
.Vec01      btst    #1,ACIA_STATUS          Interruption de l'émetteur?
            beq.s   .Vec02
            move.b  ACIA_STATUS,TxStatus(VARBASE)   Mémo état émetteur
            push    ACIA_STATUS             Sauver l'état
            bsr     TxEmpty                 -> Envoyer le prochain caractère
            pop     ACIA_STATUS             Restaure l'état
.Vec02      btst    #5,ACIA_STATUS          Overrun ?
            beq.s   .Vec03
            push    ACIA_STATUS             Sauver l'état
            bsr     OverRun                 Vider le tampon de réception
            pop     ACIA_STATUS             Restaure l'état
.Vec03:
MidiVecExit:
            popm.l  d0-d1/a0/VARBASE        Restaure registres
            rts
*
OverRun     move.b  ACIA_Data\w,d0
            rts
*
*           Interruption du transmetteur ACIA
*
TxEmpty     tst.b   TxOff(VARBASE)          SI transmission autorisée,
            bne.s   .Tx10
            move    OutHead(VARBASE),d1     SI tampon non-vide,
            cmp     OutTail(VARBASE),d1
            beq.s   .Tx10
            addq    #1,d1
            cmp     OutSize(VARBASE),d1     Wrap(Head OUT)
            bcs.s   .Tx01
            clr     d1
.Tx01       move.l  OutBuffer(VARBASE),a0   envoi du car
            move.b  0(a0,d1.w),ACIA_Data\w  courant à l'ACIA
            move    d1,OutHead(VARBASE)     update Head OUT
            bra.s   .Tx99
.Tx10       move.b  ACIACntrl(VARBASE),ACIA_Cntrl\w   Octet de ctrl
.Tx99       rts
*
*           Interruption du receveur de l'ACIA
*
RxFull      move.b  ACIA_Data\w,d0          Lecture caractère reçu
            cmpi.b  #1,XMode(VARBASE)       SI protocole XON-XOFF actif,
            bne.s   .Rx10
            cmpi.b  #XOFF,d0                XOFF reçu ?
            bne.s   .Rx01
            st      TxOff(VARBASE)          Inhibe la transmission
            bra.s   .Rx99
.Rx01       cmpi.b  #XON,d0                 XON reçu ?
            bne.s   .Rx02
            sf      TxOff(VARBASE)          Autorise la transmission
            bra.s   .Rx99
.Rx02:
.Rx10       move    InTail(VARBASE),d1      Wrap(Tail(IN))
            addq    #1,d1
            cmp     InSize(VARBASE),d1
            bcs.s   .Rx11
            clr     d1
.Rx11       cmp     InHead(VARBASE),d1      SI head=tail, tampon
            beq.s   .Rx99                   d'entrée plein, caractère perdu!
            move.l  InBuffer(VARBASE),a0
            move.b  d0,0(a0,d1.w)           caractère dans le tampon
            move    d1,InTail(VARBASE)      update Tail IN
            cmpi.b  #1,XMode(VARBASE)       SI protocole XON-XOFF
            bne.s   .Rx30
            tst.b   RxOff(VARBASE)          ET SI réception autorisée,
            bne.s   .Rx30
            sub     InHead(VARBASE),d1      ET SI tampon rempli
            bcc.s   .Rx20                   jusqu'à marée haute,
            add     InSize(VARBASE),d1
.Rx20       cmp     InHigh(VARBASE),d1
            bne.s   .Rx30
.Rx21       move.b  #XOFF,d0                envoie XOFF à l'émetteur
            bsr     MidiPut
            tst.l   d0
            bmi.s   .Rx21
            st      RxOff(VARBASE)          marque réception inhibée
.Rx30:
.Rx99       rts
*
VarBase     RSRESET
*
StndMidiVec rs.l    1
AddrMidiVec rs.l    1
*
MidiIORec   rs.w    0           IOrec du Midi (analogue RS232)
*
InBuffer    rs.l    1           Adresse du tampon d'entrée
InSize      rs.w    1           Taille du tampon d'entrée
InHead      rs.w    1           Pointeur de lecture
InTail      rs.w    1           Pointeur d'écriture
InLow       rs.w    1           Marée basse du tampon
InHigh      rs.w    1           Marée haute du tampon
*
OutBuffer   rs.l    1           Adresse du tampon de sortie
OutSize     rs.w    1           Taille du tampon de sortie
OutHead     rs.w    1           Pointeur de lecture
OutTail     rs.w    1           Pointeur d'écriture
OutLow      rs.w    1           Marée basse du tampon
OutHigh     rs.w    1           Marée haute du tampon
*
RxStatus    rs.b    1           Etat du récepteur ACIA
TxStatus    rs.b    1           Etat du transmetteur ACIA
RxOff       rs.b    1           Drapeau réception inhibée
TxOff       rs.b    1           Drapeau émission inhibée
XMode       rs.b    1           Drapeau protocole Xon-Xoff
*
MidiFlag    rs.b    1           Etat actif/inactif des routines
ACIACntrl   rs.b    1           Mémorisation ctrl pour l'ACIA
*
            dcb.b   __RS,0
*
            END
```

3615 STMAG

Téléchargement SAPRISTI.

BUTTON_X

68000

Le bouton gauche de votre souris est usé et ne fonctionne plus? Et alors? Tapez ce listing, compilez, et le tour est joué.

Michel DESSAINTES

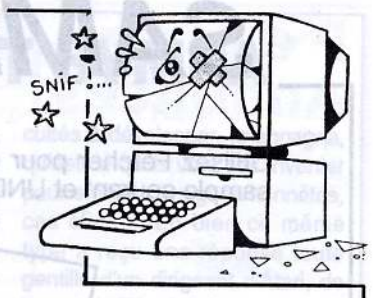
```
;
;       BUTTON_X.S : Échange les boutons de la souris.
;       ----------------------------------------------
;
; Remarques :
; ----------
;       Ce programme peut avoir une extension .TOS.
;       Il peut être exécuté au démarrage du système avec
;       la fonction INSTALLER UN APPLICATION :
;       option lancement AUTOmatique (à partir du TOS 1.4).
;
; Auteur :      Michel DESSAINTES
;
;-----------------------------------------------------------------
start_prg:      ; partie résidente
;-----------------------------------------------------------------
        bra.s   install
;-----------------------------------------------------------------
new_vec: ; échange les boutons de la souris.
;-----------------------------------------------------------------
        cmp.b   #$F9,(a0)       ; bouton gauche ?
        beq.s   inv_bit         ; oui => remplacer par le droit

        cmp.b   #$FA,(a0)       ; bouton droit ?
        bne.s   jmp_old_vec     ; non => on ne fait rien
                                ; oui => remplacer par le gauche
inv_bit:
        eori.b  #3,(a0)         ; inverser les bits
jmp_old_vec:
        jmp     $12345679       ; continuer le traitement normal
end_prg:
;-----------------------------------------------------------------
install: ; installation du programme comme résident
;-----------------------------------------------------------------
        pea     logo            ; affiche logo
        move    #9,-(sp)
        trap    #1
        addq    #6,sp
flush:                          ; flush keyboard
        move    #$FF,-(sp)
        move    #6,-(sp)
        trap    #1
        addq    #4,sp
        tst     d0
        bne.s   flush

        move    #1,-(sp)
        trap    #1              ; lire caractère du clavier
        addq    #2,sp
        or.b    #$20,d0         ; conversion en minuscules
        cmp.b   #'n',d0         ; [O]ui on [N]on ?
        beq.s   quit            ; [N]on -> quitter
        move    #$22,-(sp)              ; Kbdvbase
        trap    #14             ; XBIOS
        addq    #2,sp

        move.l  d0,a0           ; D0 -> kbdvecs
        move.l  16(a0),jmp_old_vec+2 ; vecteur souris actuel
        move.l  #new_vec,16(a0) ; changé en nouveau vecteur

        clr     -(sp)
        move.l  #end_prg-start_prg+256,-(sp)
        move    #$31,-(sp)
        trap    #1              ; fin en restant résident
quit:
        clr     -(sp)
        trap    #1              ; abandon

        section data
logo:
        dc.b    13,10,"-----------------------"
        dc.b    13,10,"Échange boutons souris"
        dc.b    13,10,"-----------------------"
        dc.b    13,10,"(c) Michel DESSAINTES   8/04/90"
        dc.b    13,10
        dc.b    13,10,"Installation : Oui/Non ? [O] ",0

        end
```

UTILITAIRE
Toutes Résolutions

POINT ASTRO

BASIC OMIKRON

Vous avez là un utilitaire qui vous permet de calculer tout ce que vous avez à calculer quand vous vous intéressez au soleil: circumzenithale, éphémérides, méridienne, Injimuvchiennidales, nbskeuxiennes, etc.



Benoit DE PRADIER

```
'*************************************************************
'**                                                         **
'**        POINT  ASTRONOMIQUE AVEC LE SOLEIL               **
'**        par Benoit de Pradier d'Agrain                   **
'**                  copyright avril 1990                   **
'**                                                         **
'*************************************************************
'
'  Sources : "Calcul astronomique pour amateurs", S. Bouiges
'
'*************************************************************
'**                    INITIALISATION                       **
'*************************************************************
'
SCREEN 1, MEMORY(32256),4'   Change d'écran graphique
PALETTE $0,,,$770'           Redéfinit la palette
DEFDBL "A-Z"'                Décimaux en double précision
DEG '                        Calculs en degrés
'
' FONCTIONS*****************************************************
'
'  Conversion degrés décimaux -> degrés, minutes
DEF FN Dms$(A#)= STR$( FIX(A#))+" degrés "+ STR$( FIX(
FRAC(A#)*60*10)/10)+" minutes "
'
'  Conversion heures décimales -> heures, minutes
DEF FN Hms$(A#)= STR$( FIX(A#))+" heures "+ STR$( FIX(
FRAC(A#)*60*10)/10)+" minutes "
'
'*************************************************************
'**                        MENU                             **
-Menu'*************************************************************
'
CLS
PRINT @(4,19);"CALCUL DU POINT ASTRONOMIQUE PAR LE SOLEIL"
PRINT @(12,10);"(1) Droite de hauteur  (2) Méridienne   (3)
Circumzénithale"
PRINT @(14,10);"(4) Point complet      (5) Ephemerides  (6)
Fin"
INPUT @(18,33);"Votre choix : ";Ch2#
CLS
ON Ch2# GOSUB Droite_De_Hauteur,Meridienne,Circumzenithale,
Point_Astro,Ephemerides,Fin
A$= INPUT$(1)
GOTO Menu
'
'*************************************************************
'**                  DROITE DE HAUTEUR                      **
-Droite_De_Hauteur'*************************************************
'
PRINT @(1,6);"DROITE DE HAUTEUR"
N#=0
GOSUB Droite
'
PRINT @(12,2);"Hauteur estimée : ";FN Dms$(He#)
PRINT @(14,2);"INTERCEPT : "; FIX(It#*10)/10;" milles "
PRINT @(15,2);"AZIMUT : ";FN Dms$(Az#);
'
RETURN
'
'*************************************************************
'**                    POINT  ASTRO                         **
-Point_Astro'******************************************************
'
'
PRINT @(1,6);"POINT ASTRONOMIQUE : deux droites de hauteur"
N#=1
GOSUB Droite
'
INPUT @(17,4);"Cap moyen du navire depuis le premier relevé en
degrés : ";Cap#
INPUT @(18,4);"Vitesse moyenne en noeuds : ";Vitesse#
'
' Calcul de l'intersection de deux droites de hauteur en
' tenant compte de la distance parcourue entre chacun des relevés
'
Temps#=H#(1)-H#(0)
B#(0)=B#(0)+ COS(Cap#-A#(0))*Vitesse#*Temps#
'
Ab#= SQR(B#(0)^2+B#(1)^2-2*B#(0)*B#(1)* COS(A#(0)-A#(1)))
Co#= SIN(A#(0)-A#(1))/Ab#
Cba#= ARCCOS(Co#*B#(0))
Cab#= ARCCOS(Co#*B#(1))
Ac#=Ab#* SIN(Cba#)/ SIN(Cba#+Cab#)
B#(2)= SQR(Ac#^2+B#(0)^2)
A#(2)=A#(0)+ ARCCOS(B#(0)/B#(2))
Lon#=Lo#+ SIN(A#(2))*B#(2)/60
Lat#=La#+ COS(A#(2))*B#(2)/60
'
PRINT @(20,2);"LATITUDE : ";FN Dms$(Lat#)
PRINT @(21,2);"LONGITUDE : ";FN Dms$(Lon#);
RETURN
'*************************************************************
'**                    MERIDIENNE                           **
-Meridienne' ******************************************************
'
PRINT @(1,6);"MERIDIENNE"
INPUT @(3,2);"Jour , mois , année : ";Jo#,Mo#,An#
INPUT @(4,2);"Longitude estimée en degrés, minutes ( - à
l'est) : ";D#,M#
Lo#= SGN(D#)*( ABS(D#)+ ABS(M#/60))
INPUT @(5,2);"Signe de la latitude estimée ( + ou - ) : ";S$
S#=(S$="-")-(S$="+")
Hd#=12
GOSUB Soleil
Hp#=Hp#+Lo#/15
PRINT @(7,4);"Heure GMT du passage du soleil au zénith : ";FN
Hms$(Hp#)
INPUT @(9,2);"Hauteur observée en degrés, minutes : ";D#,M#
Ho#=D#+M#/60
INPUT @(10,2);"Hauteur de l'oeil en mètres : ";Eo#
Hv#=Ho#-.0295* SQR(Eo#)-.016/ TAN(Ho#)+2.444E-3*
COS(Ho#)+.2667' Correction Ho
Lat#=(90-Hv#)+ ABS(De#)*S#
PRINT @(12,4);"LATITUDE : ";FN Dms$(Lat#);
RETURN
'*************************************************************
'**                    CIRCUMZENITHALE                      **
-Circumzenithale'*************************************************
'
PRINT @(1,6);"CIRCUMZENITHALE"
INPUT @(3,2);"Jour , mois , année : ";Jo#,Mo#,An#
INPUT @(5,2);"1er RELEVEMENT: Heure GMT , minutes,
secondes : ";H#,M#,S#
H1#=H#+(M#+S#/60)/60
INPUT @(6,2);"2e  RELEVEMENT: Heure GMT , minutes,
secondes : ";H#,M#,S#
H2#=H#+(M#+S#/60)/60
Hm#=(H1#+H2#)/2
Hd#=12
GOSUB Soleil
Lo#=(Hp#-Hm#)*15
PRINT @(8,4);"Temps de passage local : ";FN Hms$(Hm#)
PRINT @(9,4);"Temps de passage à Greenwich : ";FN Hms$(Hp#)
PRINT @(11,4);"LONGITUDE : ";FN Dms$(Lo#);
RETURN
'
'*************************************************************
'**                  EPHEMERIDES SOLEIL                     **
-Ephemerides'******************************************************
'
PRINT @(1,6);"EPHEMERIDES DU SOLEIL"
INPUT @(3,2);"Jour , mois , année : ";Jo#,Mo#,An#
INPUT @(4,2);"Heure GMT , minute , seconde : ";H#,M#,S#
Hd#=H#+(M#+S#/60)/60
GOSUB Soleil
PRINT @(6,2);"Ascension droite : ";FN Dms$(Ad#)
PRINT @(7,2);"Point vernal : ";FN Dms$(Pv#)
PRINT @(9,2);"DECLINAISON : ";FN Dms$(De#)
PRINT @(10,2);"A.H.G. : ";FN Dms$(Ah#)
Hd#=12
GOSUB Soleil
PRINT @(12,2);"Heure de passage du soleil à Greenwich : ";FN
Hms$(Hp#);
RETURN
'
'*************************************************************
'**          CALCUL DE LA DROITE DE HAUTEUR                 **
-Droite'***********************************************************
'
INPUT @(3,2);"Jour , mois , année : ";Jo#,Mo#,An#
INPUT @(4,2);"Latitude estimée en degrés, minutes ( - au
sud ) : ";D#,M#
La#= SGN(D#)*( ABS(D#)+ ABS(M#/60))
INPUT @(5,2);"Longitude estimée en degrés, minutes ( - à
l'est) : ";D#,M#
Lo#= SGN(D#)*( ABS(D#)+ ABS(M#/60))
FOR I#=0 TO N#
IF Ch2#=4 THEN PRINT @(7+I#*5,4);"RELEVE N°";I#+1
INPUT @(8+I#*5,2);"Heure GMT , minutes, secondes : ";H#,M#,S#
Hd#=H#+(M#+S#/60)/60
GOSUB Soleil
INPUT @(9+I#*5,2);"Hauteur observée en degrés,
minutes : ";D#,M#
Ho#=D#+M#/60
INPUT @(10+I#*5,2);"Hauteur de l'oeil en mètres : ";Eo#
Hv#=Ho#-.0295* SQR(Eo#)-.016/ TAN(Ho#)+2.444E-3* COS(Ho#)
+.2667' Correction Ho
Al#=Ah#-Lo#'                          Angle horaire local
He#= ARCSIN( SIN(La#)* SIN(De#)+ COS(La#)*
COS(De#)* COS(Al#))'                  Hauteur estimée
Az#= ARCCOS(( SIN(De#)- SIN(La#)* SIN(He#))/( COS(La#)* COS(He#)))
IF Hd#>12 THEN Az#=360-Az#'                          Azimut
It#=(Hv#-He#)*60'                                    Intercept
IF N#=1
  THEN A#(I#)=(Az#-180*(It#<0)) MOD 360'  Mémorisation des in-
formations utiles
       B#(I#)= ABS(It#)
       H#(I#)=Hd#
ENDIF
NEXT I#
RETURN
'
'*************************************************************
'**      CALCUL DES EPHEMERIDES DU SOLEIL                   **
-Soleil'***********************************************************
'
' T=temps écoulé en jour depuis le 1er janvier 1901 à 0 heure
T#= FIX((An#-1901)*365.25+ FIX(((Mo#-1)*30.59)+.45)+1.75
*(Mo#>2)+Jo#)+Hd#/24
J$="LUNDI   MARDI   MERCREDIJEUDI   VENDREDISAMEDI   DIMANCHE"
PRINT @(3,50);"Nous sommes "; MID$(J$,( FIX(T#) MOD 7)*8+1,8)
Ob#=23.439'                           Obliquité de l'écliptique
Ex#=.016713416#'                      Exentricité
Lm#=(278.9654+.9856473421#*T#) MOD 360'    Longitude moyenne
Pv#=(Lm#+180+15*Hd#) MOD 360'         Point vernal
Lp#=281.235+4.69E-5*T#'               Longitude du périhélie
Am#=Lm#-Lp#'                          Anomalie moyenne
'
U#=Am#
FOR K#=1 TO 10'                       Equation de Kepler
  U#=Am#+Ex#* SIN(U#)*180/ PI
NEXT K#
'
Av#=2* ATN( SQR((1+Ex#)/(1-Ex#))* TAN(U#/2))'  Anomalie vraie
Lv#=(Lp#+Av#) MOD 360'                         Longitude vraie
'
De#= ARCSIN( SIN(Lv#)* SIN(Ob#))'              Déclinaison
Ad#= ARCCOS( COS(Lv#)/ COS(De#))'              Ascension droite
IF COS(Ob#)* SIN(Lv#)<0 THEN Ad#=360-Ad#'      Angle complémentaire
Ah#=(Pv#-Ad#+360) MOD 360'                     AHvo
IF Hd#=12 THEN Hp#=(12+(360-Ah#)/15) MOD 24'   Temps de passa-
ge au méridien 0
RETURN
'
'*************************************************************
'**                        FIN                              **
-Fin'**************************************************************
'
PALETTE $777
END
```

ASTRONOMIE
Toutes Résolutions

il a tout pour vous plaire!

DISQUE DUR Golden Premium
GARANTIE 2 ANS

A PARTIR DE 3750 F TTC

RAPIDE ET FACILE D'EMPLOI / SILENCIEUX

Code produit	Débit (en Ko/s)	Temps d'accès (en ms)	Capacité formatée (en Mo)	Prix TTC
GP 40	680	19	40	3750.00
GP 52	1000	17	52	3990.00
GP 105	1000	17	105	5990.00
GP 210	1000	15	210	8990.00

-Mémoire cache de 64 Ko intégrée.
-Logiciels de formatage et de partitionnement fournis.
-Disque auto-parquable.
-Horloge permanente intégrée.
-Taille ultra réduite (15 l x 30 L x 4.5 H)
-Disque dur Quantum.
-Sortie SCSI

LIVRE PRET A FONCTIONNER

Nouveau
-Protection en écriture.
-Extinction automatique au bout de 30 min.

100% COMPATIBLE DISQUE ATARI

et aussi...

-Extension mémoire Big Mem pour Mega ST1/STf (2 ou 4 Mo).
490.F00 - Code BIG MEM.
-Extension mémoire pour STe (barrette SIM).
350.F00 - Code SIM.
-Extension mémoire 512 Ko pour 520 STf.
320.F00 - Code Ext 520.
-Kit haute densité pour STf/STe/Mega (sans le lecteur).
290.F00 - Code KITHD.
-Lecteur haute densité interne.
495.F00 - Code LECHDI.

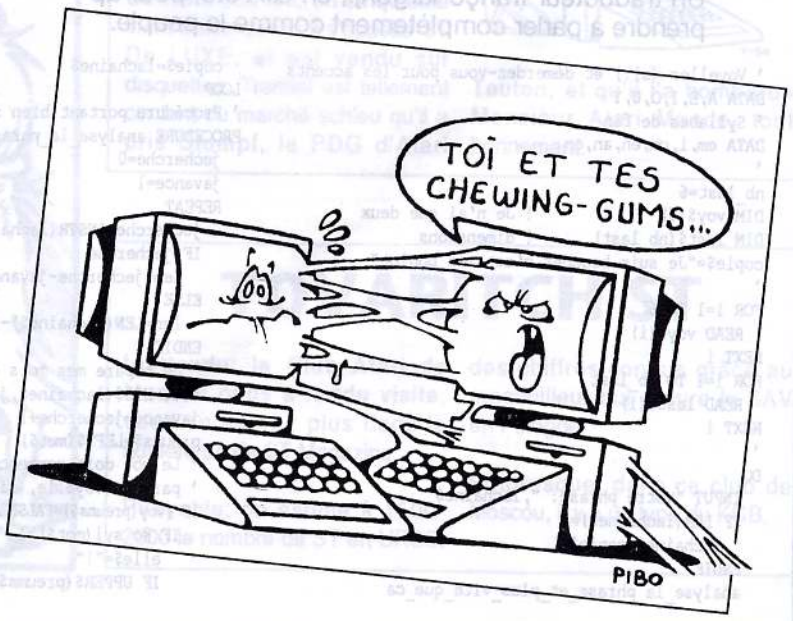
Bon de Commande
Remplissez lisiblement le bon ci-dessous et adressez le à:
OMIKRON, 7 rue VOLTAIRE - 51100 REIMS
Tél. 26 40 60 22 - Fax. 26 97 71 39

nom : ____ prénom : ____
adresse : ____
code postal : ____ ville : ____
je règle par : ☐ chèque bancaire ☐ contre remboursement: 60F ☐ carte bleue
n° ____
date de validité ____

Article	Quantité	Référence	Total
		Port disque dur: 100 F	
		Port autres produits: 39 F	
		Contre remboursement: 60 F	
		Total à régler	

PAYEZ EN 4 FOIS / SATISFAIT OU REMBOURSE
Consultez nous pour plus d'information.

OMIKRON
OMIKRON, 7 rue VOLTAIRE - 51100 REIMS
Tél. 26 40 60 22 - Fax. 26 97 71 39
LIVRAISON RAPIDE / CHRONOPOST



SAMPLES FETCHER

GFA-BASIC

Utilisez Fetcher pour écouter et tripoter des samples (.SPL). HELP garde en mémoire le sample courant et UNDO le restitue.

MUSIQUE
Basse Résolution

ULTIMATUM
of Zap Creation (from The Heavy Killers)

```
HIDEM
FOR a%=0 TO 191
  READ b$
  c$=c$+MKL$(VAL("&H"+b$))
NEXT a%
FOR a%=1 TO 768 STEP 3
  tb$=tb$+CHR$(8)+MID$(c$,a%,1)+CHR$(9)+MID$(c$,
SUCC(a%),1)+CHR$(10)+MID$(c$,a%+2,1)+MKI$(0)
NEXT a%
FOR a%=0 TO 31
  READ b$
  play$=play$+MKL$(VAL("&h"+b$))
NEXT a%
speed%=46
SPOKE &H484,0
CLR c$
SETCOLOR 0,0
SETCOLOR 15,&H765
SETCOLOR 1,&H753
SETCOLOR 2,&H321
SETCOLOR 3,&H357
SETCOLOR 4,&H567
SETCOLOR 5,&H725
SETCOLOR 7,&H700
CLS
COLOR 3
BOX 0,0,319,127
COLOR 2
LINE 1,64,318,64
step%=10
cut%=2
aftxt
CLR s$
DO
  IF s$=""
    KEYPRESS 187
  ENDIF
  SELECT SUB(INP(2),186)
  CASE 1
    ALERT 2," | |   Format ? | ",ADD(2,format%),
" Amiga | Atari ",stmag%
    format%=SUB(stmag%,2)
    FILESELECT "A:\*.SPL",MID$(nf$,4),nf$
    HIDEM
    IF EXIST(nf$)
      OPEN "I",#1,nf$
      ls%=MIN(LOF(#1),32767)
      s$=INPUT$(ls%,#1)
      CLOSE #1
      IF format%
        FOR g%=V:s$ TO PRED(ADD(V:s$,ls%))
          BYTE{g%}=BCHG(BYTE{g%},7)
        NEXT g%
      ENDIF
      undo$=s$
      unls%=ls%
      redraw
    ENDIF
  CASE 2
    ALERT 2," | |   Format ? | ",ADD(2,format%),
" Amiga | Atari ",stmag%
    format%=SUB(stmag%,2)
    FILESELECT "A:\*.SPL",MID$(nf$,4),nf$
    HIDEM
    IF format%
      sv$=s$
      FOR g%=V:s$ TO PRED(ADD(V:s$,ls%))
        BYTE{g%}=BCHG(BYTE{g%},7)
      NEXT g%
    ENDIF
    IF nf$<>""
      BSAVE nf$,V:s$,ls%
    ENDIF
    IF format%
      s$=sv$
      CLR sv$
    ENDIF
  CASE 3
    boost:
    CLR mx%,mn%
    SETCOLOR 0,4
    FOR ad%=V:s$ TO ADD(PRED(ls%),V:s$)
      k%=SUB(BYTE{ad%},128)
      mx%=MAX(k%,mx%)
      mn%=MIN(k%,mn%)
    NEXT ad%
    IF mn%<=-128 OR mx%>=127
      SETCOLOR 0,0
      ALERT 3," | | Unboostable | ",1,
" Sorry ",kjkj$
      HIDEM
    ELSE
      ar%=MAX(mx%,-mn%)
      FOR x&=0 TO 319
        PSET x&,SHR(SUB(128,ar%),1),ADD(PTST(x&,SHR
(SUB(128,ar%),1)),4)
        PSET x&,SHR(ADD(127,ar%),1),ADD(PTST(x&,SHR
(ADD(127,ar%),1)),4)
      NEXT x&
      rap=128/ar%
      FOR ad%=V:s$ TO ADD(PRED(ls%),V:s$)
        BYTE{ad%}=MIN(ADD(128,SUB(BYTE{ad%},
128)*rap),255)
      NEXT ad%
      SETCOLOR 0,0
      redraw
    ENDIF
  CASE 4
    FOR ad%=V:s$ TO ADD(PRED(ls%),V:s$)
      EXIT IF ABS(SUB(BYTE{ad%},128))>cut%
    NEXT ad%
    start%=SUCC(ad%-V:s$)
    FOR ad%=ADD(PRED(ls%),V:s$) DOWNTO V:s$
      EXIT IF ABS(SUB(BYTE{ad%},128))>cut%
    NEXT ad%
    os%=ls%
    ls%=SUCC(SUB(SUCC(ad%-V:s$),start%))
    IF ls%>0
      s$=MID$(s$,start%,ls%)
      redraw
    ELSE
      ls%=os%
    ENDIF
  CASE 5
    GRAPHMODE 3
    WHILE MOUSEK
    WEND
    or%=-1
    COLOR 5
    REPEAT
      MOUSE mx%,my%,mz%
      IF my%>128
        SETMOUSE mx%,128
      ENDIF
      ar%=MIN(64,MAX(1,ABS(64-my%)))
      IF ar%<>or%
        LINE 0,ADD(63,ar%),319,ADD(63,ar%)
        LINE 0,SUB(64,ar%),319,SUB(64,ar%)
        IF or%<>-1
          LINE 0,ADD(63,or%),319,ADD(63,or%)
          LINE 0,SUB(64,or%),319,SUB(64,or%)
        ENDIF
        or%=ar%
      ENDIF
    UNTIL mz%
    LINE 0,ADD(63,ar%),319,ADD(63,ar%)
    LINE 0,SUB(64,ar%),319,SUB(64,ar%)
    GRAPHMODE 1
    SETCOLOR 0,4
    ADD ar%,ar%
    lim%=ar%
    CLR mx%,mn%
    SETCOLOR 0,4
    FOR ad%=V:s$ TO ADD(PRED(ls%),V:s$)
      k%=SUB(BYTE{ad%},128)
      mx%=MAX(k%,mx%)
      mn%=MIN(k%,mn%)
    NEXT ad%
    ar%=MAX(mx%,-mn%)
    FOR x&=0 TO 319
      PSET x&,SHR(SUB(128,ar%),1),ADD(PTST(x&,SHR
(SUB(128,ar%),1)),4)
      PSET x&,SHR(ADD(127,ar%),1),ADD(PTST(x&,SHR
(ADD(127,ar%),1)),4)
    NEXT x&
    rap=lim%/ar%
    FOR ad%=V:s$ TO ADD(PRED(ls%),V:s$)
      BYTE{ad%}=MIN(ADD(128,SUB(BYTE{ad%},
128)*rap),255)
    NEXT ad%
    SETCOLOR 0,0
    redraw
  CASE 6
    SETCOLOR 0,4
    FOR ad%=V:s$ TO ADD(PRED(ls%),V:s$)
      IF BYTE{ad%}>ADD(128,cut%)
        BYTE{ad%}=SUB(BYTE{ad%},cut%)
      ELSE IF BYTE{ad%}<SUB(128,cut%)
        BYTE{ad%}=ADD(BYTE{ad%},cut%)
      ELSE
        BYTE{ad%}=128
      ENDIF
    NEXT ad%
    GOTO boost
  CASE 7
    SGET z$
    PRINT AT(1,17);
    INPUT "Step=1/",hg$
    IF VAL(hg$)
      step%=VAL(hg$)
    ENDIF
    SPUT z$
    CLR z$
    aftxt
    redraw
  CASE 8
    SGET z$
    PRINT AT(1,17);
    INPUT "Cut under ",hg$
    IF hg$<>""
      cut%=MIN(ABS(VAL(hg$)),127)
    ENDIF
    SPUT z$
    aftxt
    redraw
  CASE 9
    PRINT AT(1,17);
    INPUT "Speed=",speed%
    speed%=ABS(speed%)
    PRINT AT(1,17);STRING$(40,32);
    PRINT AT(16,17);CHR$(27);"b1Len:";ls%''''AT
(27,17);"Speed:";speed%;CHR$(27);"b?";
  CASE 10
    OUT 4,&H12
    super%=GEMDOS(&H20,L:0)
    play%=V:play$
    tb%=V:tb$
    REPEAT
      ~C:play%(L:V:s$,L:V:s$+ls%,L:tb%,speed%)
    UNTIL BYTE{&HFFFC02}<>68
    ~GEMDOS(&H20,L:super%)
    OUT 4,8
  CASE 39
    s$=undo$
    ls%=unls%
    redraw
  CASE 40
    undo$=s$
    unls%=ls%
  CASE -159
    SPOKE &H484,7
    EDIT
  ENDSELECT
LOOP
PROCEDURE aftxt
  PRINT AT(1,18);"SAMPLES FETCHER v1.3 coded for
ST-Mag by";
  PRINT AT(1,19);"  Ultimatum of Zap Creation from
T.H.K";
  PRINT AT(5,20);STRING$(32,255);
  PRINT AT(1,21);"[F1]:Load Sample";AT(20,21);
"[F2]:Save Sample";
  PRINT AT(1,22);"[F3]:Volume Booster[F4]:
Blank killer";
  PRINT AT(1,23);"[F5]:Set Volume    [F6]:Low-Pass
Filter";
  PRINT AT(1,24);"[F7]:Step=1/";step%''''''AT(20,
24);"[F8]:Cut at ";cut%''
  PRINT AT(1,25);"[F9]:Replay speed";AT(20,25);
"[F10]Replay";
RETURN
PROCEDURE redraw
  PRINT AT(16,17);CHR$(27);"b1Len:";ls%''''AT(27,
17);"Speed:";speed%;CHR$(27);"b?";
  rap=ls%/320
  DEFFILL 0
  PBOX 0,0,319,127
  COLOR 3
  BOX 0,0,319,127
  COLOR 2
  LINE 1,64,318,64
  FOR x=0 TO 319 STEP 1/step%
    PSET x,SHR(BYTE{V:s$+x*rap},1),3
  NEXT x
  FOR x&=0 TO 319
    PSET x&,SHR(SUB(128,cut%),1),ADD(PTST(x&,SHR
(SUB(128,cut%),1)),2)
    PSET x&,SHR(ADD(128,cut%),1),ADD(PTST(x&,SHR
(ADD(128,cut%),1)),2)
  NEXT x&
  LINE 1,64,318,64
RETURN
DATA C0B090C,B090D08,80B0B0B,D09050C,B080D09
DATA 20D0806,C0B070D,7070C0B,60C0A09,B0B0A0C
DATA B020C0B,C0A08,D06040D,5050D05,40C0909,D04030B
DATA B090C0A,50B0A0A,C09080B,B080C0A,C0A00
DATA C09070B,B070C09,60B0B06,B0A090B,B050A0A
DATA A0B0B02,B0A080C,7070C08,40C0706,B09090C,6060A0A
DATA 90C0703,B0A050B,9080B0A,30A0A08,B0A000B
DATA 9070B08,80A0A07,A09090C,1010A0A,60B0807
DATA A0A050A,9080A0A,20A0A01,A0A0009,9090A08,80B0801
DATA A09060B,7040A09,5090908,A09030A,8060A09
DATA 90907,908080A,8040909,60A0801,9090509,8070808
DATA 8090902,9080609,9000907,7080807,9070609
DATA 8020808,6090606,8070708,8040807,6080802
DATA 7070708,6060807,4070706,8060508,6040706
DATA 6070704,8050406,6060706,4070505,6060506,6040605
DATA 5060602,6050405,5050605,2050504,5040405
DATA 5020404,4040403,4040204,3030303,3030302
DATA 3020202,2020202,1010101,2010001,1000100,0
DATA E0D0C0F,3000F03,F0300,F03000F,3000F03,E0D0B
DATA E0D0B0E,D0B0E0D,B0E0D0B,E0D0B0E,D0B0E0D
DATA A0E0D0A,E0D0A0E,D0A0E0C,C0E0D00,D0D0D0D
DATA D0D0D0D,D0D0D0D,D0D0D0D,D0D0E0C,B0E0C0B
DATA E0C0B0E,C0B0E0C,B0E0C0B,E0C0B0E,C0B0E0C
DATA A0E0C0A,E0C0A0E,C0A0D0D,C0D0D0C,E0C090E,C090E0C
DATA 50E0C00,E0C000E,B0B0E0B,B0E0B0B,E0B0B0E
DATA B0A0E0B,A0E0B0A,D0D0B0D,D0B0D0D,B0E0B09
DATA E0B090E,B090D0C,C0D0D0A,E0B070E,B000E0B,D0D09
DATA D0D090E,A090D0D,80D0D07,D0D040D,D000E0A
DATA 40E0909,E09090D,C0B0E09,80E0908,E09070E
DATA 8080E09,10C0C0C,D0C0A0E,8060E07,70E0800,E07050E
DATA 6060D0C,90E0505,E04040D,C080D0B,B0E0000
DATA D0C060D,C050D0C,20C0C0B,C0C0B0D,B0A0D0B
DATA A0D0B0A,D0B0A0C,C0A0C0C,A0C0C0A,D0B090D,B090D0A
DATA A0D0A0A,D0A0A0C,C090C0C,90C0C09,D0B060C
DATA B0B0C0C,80D0B00,D0B000C,C070C0C,60C0C05
DATA C0C030C,C010C0B,A0D0A05,D0A040D,A020D09,80D0908
DATA 46FC2700,48E7F0F0,206F0024,226F0028,246F002C
DATA 302F0030,343C0080,47F88800,760A1683,C030007
DATA 670651EB,26004,50EB0002,51CBFFEC,72001218,D202E749
DATA 26321000,32321004,7CB0000,38B0000,360051CB
DATA FFFEB3C8,66DE203C,8000900,1CB0000,303C0A00
DATA 18B0000,46FC2300,4CDF0F0F,4E757274,2E206279
DATA 2054484B ! Comme aù bon vieux temps!..
```

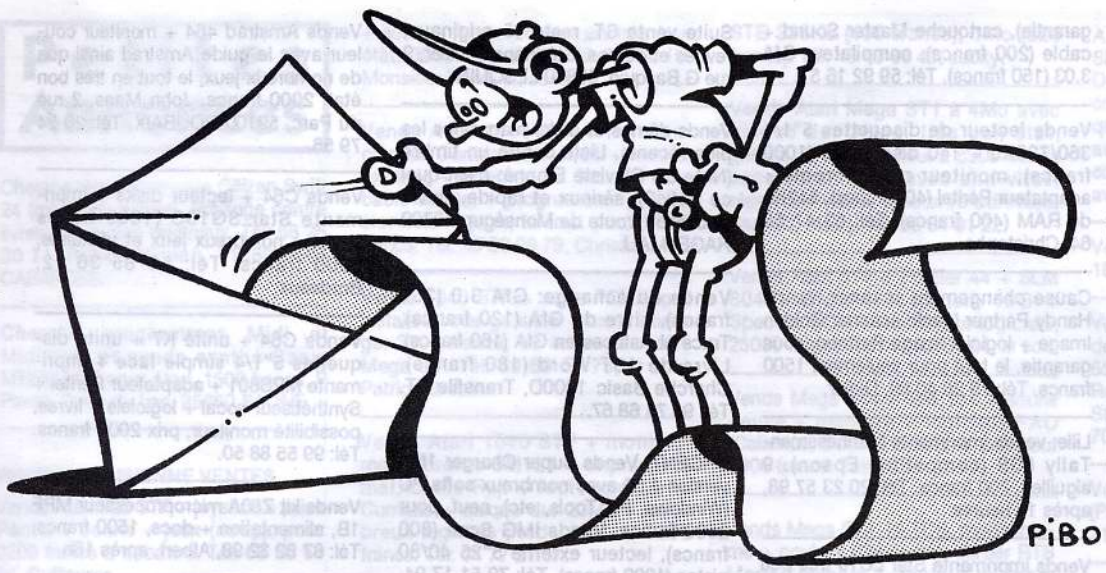
LARGONJI

LASICBEM GFA

Un traducteur franço-largonji, en GfA 3.0, pour apprendre à parler complètement comme le peuple.

Mic DAX

```
' Voyelles (si.) et démerdez-vous pour les accents
DATA A,E,I,O,U,Y
' Syllabes de fin
DATA em,i,om,en,an,on
'
nb_last=6
DIM voy$(6)          ! Je n'ai que deux
DIM last$(nb_last)   ! dimensions
copie$="Je suis boucher c'est mon boulot"
'
FOR i=1 TO 6
  READ voy$(i)
NEXT i
FOR i=1 TO nb_last
  READ last$(i)
NEXT i
'
DO
  INPUT "Votre phrase: ",lachaine$
  IF LEN(lachaine$)=0
    lachaine$=copie$
  ENDIF
  analyse_la_phrase_et_plus_vite_que_ca
  copie$=lachaine$
LOOP
' Procédure portant bien son nom
PROCEDURE analyse_la_phrase_et_plus_vite_que_ca
  jecherche=0
  javance=1
  REPEAT
    jecherche=INSTR(lachaine$," ",javance)
    IF jecherche
      len=jecherche-javance
    ELSE
      len=LEN(lachaine$)-javance+1
    ENDIF
    ' Je sépare mes mots comme il faut...
    mot$=MID$(lachaine$,javance,len)
    javance=jecherche+1
    preums$=LEFT$(mot$)
    ' Le mot doit commencer par une consonne puis
    ' par une voyelle, sinon c'est n'importe quoi.
    IF @voy(preums$)=FALSE AND @voy(MID$(mot$,2,1))=TRUE
      IF @c_syl(mot$)>1
        elle$="l"
        IF UPPER$(preums$)=preums$
          elle$="L"
        ENDIF
        ' On tire la dernière syllabe au hasard
        mot$=elle$+MID$(@arrange$(mot$),2)+
@lower$(preums$)+last$(RND*nb_last+1)
      ENDIF
    ENDIF
    PRINT mot$;" ";
  UNTIL jecherche=0
  PRINT
RETURN
' Avec de si beaux yeux, tu es au moins une voyelle!
FUNCTION voy(car$)
  LOCAL i
  FOR i=1 TO 6
    IF UPPER$(car$)=UPPER$(voy$(i))
      RETURN TRUE
    ENDIF
  NEXT i
  RETURN FALSE
ENDFUNC
' Il est à toi, ce petit sac?
' Fais voir combien il y a de syllabes dedans?
FUNCTION c_syl(m$)
  LOCAL c$
  syllabes=0
  voyelle=FALSE
  FOR i=2 TO LEN(m$)
    c$=MID$(m$,i,1)
    ' Je compte les syllabes
    IF @voy(c$)=TRUE
      IF voyelle=FALSE
        INC syllabes
        voyelle=TRUE
      ENDIF
    ELSE
      voyelle=FALSE
    ENDIF
  NEXT i
  RETURN syllabes
ENDFUNC
' Joue pas les grandes avec moi,
' T'es rien qu'une minuscule.
FUNCTION lower$(m$)
  IF ASC(m$)>64 AND ASC(m$)<91
    m$=CHR$(ASC(m$)+32)
  ENDIF
  RETURN m$
ENDFUNC
FUNCTION arrange$(m$)
  LOCAL c$
  FOR i=LEN(m$) DOWNTO 1
    c$=MID$(m$,i,1)
    EXIT IF @voy(c$)=TRUE
  NEXT i
  RETURN LEFT$(m$,i)
ENDFUNC
```



COURRIER DES LECTEUX

Votre équipe de rédaction, et surtout, "la Voix de son Maître", je veux dire la SARL Pressimage, auront-elles le courage de publier la présente dans le Courrier des Lecteurs?

Mr BESNEUX, CLICHY (92)

Non. D'autres questions?

Pressimage est peut-être indépendant... mais ST Mag, lui, n'est-il pas (un peu trop!) étroitement lié à Pressimage, qui n'est jamais, après tout, comme tout éditeur qui se respecte, qu'une machine à gagner des sous, fût-ce au détriment des lecteurs, pourvu que sa Sainteté (sale) Fric rentre, n'est-ce-pas?
La lecture d'un extrait K-Bis du registre du Commerce serait sans doute pleine de renseignements révélateurs, tels qu'une récente participation (ou augmentation de participation) substantielle de Pressimage dans le capital et le Conseil de Direction de ST Mag, même si, à l'origine, sans doute, ST Mag était totalement indépendant de Pressimage, alors simple prestataire de services.
Cette prise de contrôle a dû provoquer de sérieux remous dans la direction et la Rédaction, au premier semestre, non? Les commentaires "partira-partira pas" dans le cadre en dessous de l'édito du numéro 48 de Janvier sont très révélateurs à cet égard, plus d'autres signes avant-coureurs qui m'inquiétaient.
Ainsi, mes craintes étaient fondées, le canard replie ses ailes, frileusement, sans doute aux ordres de Sa Majesté Pressimage, au lieu de continuer à s'ouvrir largement à l'information sur l'informatique, sans pour autant délaisser le ST.

Mr BESNEUX, CLICHY (92)

C'est pour la caméra cachée, c'est un gag? Monsieur BESNEUX, vous regardez trop les feuilletons américains et vous ne lisez pas assez notre modeste magazine. Pressimage est la société qui édite ST Magazine, qui n'est qu'un canard, pas une sociéte. De fait, ST Mag n'a aucun capital qui lui soit propre, et encore moins de Conseil de Direction. Notre Directeur de la Publication est le patron de Pressimage, il a décidé il y a maintenant six ans de créer un journal dédié au ST à partir de l'entreprise qu'il avait créée pour éditer un magazine déjà dédié au standard Atari. Il n'y a donc aucune prise de participation, aucune OPA, aucun noyau dur, aucune pression, puisque les liens entre ST Magazine et Pressimage sont on ne peut plus naturels. Vous avez raison sur un point, c'est heureux, une entreprise ne peut pas survivre si elle ne fait pas de profits. Bravo.

Quant au "partira-partira pas", il s'agissait d'une référence à la date du 15 janvier, date de l'ultimatum posée à l'Irak par les Etats-Unis, la question était de savoir quels seraient ceux d'entre nous qui iraient combattre en cas de conflit. C'était juste une plaisanterie, veuillez nous en excuser.

Avez-vous, en prenant cette funeste décision de fermeture des frontières (NDLR: notre lecteur clair comme de l'eau de roche veut parler de notre retour à la formule 100% ST) pensé à la conséquence? Le retrait d'agrément de la Commission Paritaire, agrément que vous aviez mis des années à obtenir, vous pend au nez. Et qui va payer les pots cassés? Le client, le "cher lecteur", le connard, quoi!

Mr BESNEUX, CLICHY (92)

Oui, oui, promis, nous y avions pensé, ça nous arrive de réfléchir, parole d'honneur.
Nous espérons pouvoir conserver notre numéro de Commission Paritaire encore quelque temps, mais il est vrai que nous avons fait une croix dessus, et figurez-vous que nous avons tout bien pesé, le pour, et aussi le contre, et encore le pour, et à nouveau le contre, pour être bien sûrs qu'on ne se tromperait pas.
Notre agrément était de toutes façons provisoire et reconduit après de sévères contrôles, la Commission nous interdisant par exemple de citer les coordonnées des éditeurs et le prix des produits que nous vous présentions, interdiction qui déroutait bon nombre de nos lecteurs tombant sur un canard n'ayant pas le droit de les informer. Nous tâcherons de ne pas vous importuner avec nos déboires admnistratifs, nous ne prenons pas nos lecteurs pour ce que vous dites; naïfs que nous sommes, nous pensions que les efforts faits dès le numéro 53 démontraient notre respect envers le lecteur.

ST Magiciel. Première critique: c'est sans doute encore une de ces idées de Pressimage pour pomper nos pauvres poches des maigres sous qui leur restent... (tant pis si cet éditeur en prend pour son grade ici, il fallait bien que quelqu'un le remette un peu à sa place: il est et doit rester un IMPRIMEUR, et non un dictateur dictant ses quatre caprices à son client).

Mr BESNEUX, CLICHY (92)

C'est reparti comme en 40. Pressimage n'a rien imposé du tout, l'idée de ce hors-série vient de la Rédaction de ST Magazine, et notre Directeur de la Publication (aussi patron de l'Infâme Pressimage) a dit "fonce, coco", et aussi "accessoirement, ne fais pas n'importe quoi avec les sous de Pressimage, ils vont encore nous engueuler".

Deuxième critique: ce n'est pas une heureuse idée, beaucoup d'entre nous, j'en suis certain, n'apprécieront pas du tout cette séparation: pourquoi ne pas avoir maintenu l'ancienne formule, au besoin en regroupant tous les programmes dans un encart dans un cahier intérieur?

Mr BESNEUX, CLICHY (92)

Nous nous sommes mal compris: ST Magazine continuera à publier des listings dans ses pages, nous avions juste envie, depuis quelques mois, de sortir un supplément avec beaucoup beaucoup de listings, une grande partie de nos lecteurs nous en demandant plus, toujours plus. Il était impossible de les intégrer tous dans ST Mag, déjà bien rempli par des articles de fond. On vous demande de l'argent, oui, pour pouvoir partir en vacances, et pour que le Directeur De La Publication puisse créer sa propre boite et se libérer du joug de Pressimage, dont il est le patron.

Troisième critique, la plus grave (NDC: mamma mia, j'ai peur!): j'ai trouvé en page 131 la présentation du futur STMAGICIEL, et son aspect m'a rappelé, bien sûr (NDC: bien sûr), celui d'un hebdomaire célèbre pour la vulgarité de son style ordurier, et qui a bu le "bouillon" en grande partie à cause de cela...
Prendriez-vous vos lecteurs pour des cons, pour employer le style du dit "canard" boiteux?
Je crains fort que dans cette affaire, si vous employez un tel style, ce soit vous les c... , au bout du compte, et votre canard bis risque alors de faire un "bide".
Le genre loubard n'est pas si fréquent que vous pourriez le croire parmi vos lecteurs.
Or j'ai lu votre présentation à la loupe, et ce que j'y ai lu ne fait que confirmer mes craintes: vous déraillez, les pôvres, et vous allez prendre un sérieux coup de b..., pardon de pied au c..., si vous persistez dans ce style vaseux.

Mr BESNEUX, CLICHY (92)

Vous me rappelez ces gens qui se forcent à aller dans les sex-shops pour pouvoir dire de visu que c'est abominable, honteux et que l'Occident est en pleine décadence. Lire à la loupe des textes "choquants", il faut le vouloir, personne ne vous y oblige, jetez votre loupe et qu'on n'en parle plus, bon sang. Nous ne cherchons pas adopter un style "loubard", seulement nous préférons de loin pouvoir dire "bite", "cul" et "con", plutôt que d'avoir à nous contorsionner en nous exprimant à l'aide de points de suspension, ça fait moins "balai dans le c...". Quel dommage que vous ne puissiez lire notre réponse aussi complète que précise, quoiqu'il serait amusant que vous vous soyez fait violence et que vous vous obligeâtes à vous procurer notre hors-série.
Par ailleurs, nous sommes bien heureux d'apprendre qu'Hebdogiciel avait pris la soupe, nous avions une autre version. Voir page 36.

Pourriez-vous arrêter de vous foutre de la gueule des Bretons, s'il-vous-plait, nous ne sommes pas plus cons que les autres. Merci de tout coeur. KENAVO.

Pascal LE DELLIOU Orléans (45)

Désolé, mais des gens qui disent "Breizenec ken kerlesclengen har in Plouay ant viva trugarez", qui mangent des galettes saucisses et qui se font reléguer en Seconde Division, désolé, non, c'est trop risible. Muchisimos besos, y hasta luego!

SAM EST SYMPA

On le croyait sur son nuage, entouré des plus belles nanas du monde, se prélassant autour de sa piscine en forme du logo d'Atari, mais c'est un altruiste, notre Tramiel.

Un type lui a écrit pour se plaindre, des histoires de difficultés à développer, gnagnagna, qu'est-ce-qu'ils vont pas inventer pour embêter les gens honnêtes, ces clients. Eh bien ce même type a reçu une réponse toute gentille d'un dirigeant d'Atari, de la part de Tramiel, étonnant non? Il aurait pu lui offrir un TT, ça aurait été un beau conte de fées.

PERIL JAUNE

Tiens, vous savez pourquoi les machines d'Atari sortent toujours en retard? Parce qu'elles avaient du mal à passer les tests du FCC (Federal Communications Commission, qui traque les interférences radio chez tout appareil électrique lui passant dans les pattes). Et pourquoi elles avaient du mal à passer ces tests? Parce qu'à chaque coup ou presque, il y avait un ingénieur d'Atari Japon qui s'amusait à "améliorer" la machine qu'on lui demandait de fabriquer. Résultat, Atari Corp a dépêché un des siens, Jim "Eastwood" Tittsler, pour faire respecter les décisions prises aux USA, non mais des fois.

UN PARTOUT

Tout le monde dit que le ST est mort. La preuve, START, ce canard ricain consacré au ST, START a passé l'arme à gauche.

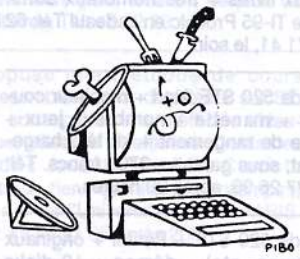
D'un autre côté, des utilisateurs du serveur GENIE (toujours aux States) s'apprêtent à lancer un canard dédié au ST.

C'EST L'HISTOIRE D'UN ATARI KIBBOUTZ

Elie Kenan, c'est l'ex-PDG d'Atari France, il a laissé son poste fin 90 pour s'occuper d'Atari Amérique Du Nord And Canada. Il paraitrait qu'il n'a jamais obtenu son visa, des histoires de maladie, on sait pas. Heureusement, un bruit court comme quoi il ouvrirait une usine Atari en Israël, avec une subvention du gouvernement hébreu à la clé. Un autre bruit court et prétend que l'usine serait implantée dans les territoires occupés, avec une subvention encore plus grosse que ça.
Depuis l'été 89, on sait qu'Atari résiste mal au feu, Kenan "j'ai testé pour vous" veut à présent tâter du caillou.
Bientôt sur vos écrans: Atari Irlande du Nord, Atari Philippines, Atari Corse, Atari Pays Basque, et Atari Kurdistan.

ST UBER ALLES

En Allemagne, le ST se porte plutôt bien, quatre canards lui sont consacrés: ST Computer, ST Magazin, TOS Magazin et PD Mag (plus tourné vers les softs du Domaine Public sur ST). Ce n'est pas fini, un cinquième vient d'être lancé, il s'appelle ST De LUXE, et est vendu sur disquettes. Tramiel est tellement content du marché schleu qu'il a pris Stumpf, le PDG d'Atari Teuton, et qu'il l'a bombardé Monsieur Atari Monde, tout bonnement.



TOVARITCH ST

Inattendu: le Club Atari de Moscou nous a rendu visite, vous en saurez plus dans le numéro 54 de ST Magazine.

Pitoyable: on estime à 1500-2000 le nombre de ST en URSS, des chiffres connus grâce au magouilleur qui assure le SAV en Pologne.

Pittoresque: dans ce club de Moscou, il y a un type du KGB.

Petites Annonces gratuites

Mode d'emploi. Lisez d'abord une petite annonce, n'importe laquelle, au hasard, pour voir un peu le genre, et revenez, je vous explique ensuite. Voilà, dans une petite annonce, vous avez trois parties. Si ce n'est pas le cas de celle que vous avez lue, choisissez-en une autre, vous n'avez pas eu de chance. Première partie, on vous présente l'objet de la vente ou de l'achat, avec plus ou moins de détails. Ne vous laissez pas embobiner par des arguments trop faciles, genre "jamais servi"; vous avez déjà connu ça, vous, un type qui ne se sert pas d'un ordinateur d'un demi-bâton? Non, vous voyez bien. Bon, ensuite, pour une vente, on vous indique combien l'auteur de la PA en veut, en francs français. C'est parfois à débattre, c'est à dire que vous avez le droit de négocier, vous vous en tirerez avec une ristourne de 200 ou 300 francs et vous aurez le droit de penser que vous êtes un type à qui on ne la fait pas, des marchandeurs comme vous on n'en fait plus. Et pour contacter le type qui a passé l'annonce, c'est la troisième partie, on vous donne ses coordonnées, c'est ça qui est super avec les petites annonces, on se fait plein d'amis. On décroche son téléphone, on compose le numéro, et on dit "Bonjour, c'est pour l'annonce que j'appelle / mon nom à moi c'est Marcel / Je suis un rappeur fantastique / et de ton matos j'en veux bernique / tu peux te le carrer dans l'agneau / mon gars." et on raccroche aussi sec. Bon, non, c'est pas comme ça, il faut appeler aux heures qui sont données, selon le principe "avant l'heure c'est pas l'heure après l'heure c'est plus l'heure", et demander la personne dont le prénom est parfois donné. Avec ça, si vous n'y arrivez pas, c'est que vous êtes vraiment trop ahuri, c'est bien la dernière fois que j'explique un truc aussi con, on m'avait bien conseillé de choisir un autre boulot mais alors là c'est un peu fort. Bon, ciao, bonne chance, et ne répondez pas aux chaines, c'est salades et compagnie.

Si vous voulez passer une PA à votre tour, renvoyez le bon inséré dans ST Magazine, ça coûte zéro franc zéro centime, écrivez proprement et rien que des choses honnêtes, parce que les pattes de mouche, les bulletins raturés, avec 3 couches de Typex et les annonces pour copies de softs piratés, vous pouvez vous les rouler en pointe, ça passe pas, il y a suffisamment d'annonces honnêtes et bien écrites pour qu'on se les casse pas à déchiffrer les petits tas d'encre, merci, soyez sympas avec le petit personnel.

PROVINCE VENTES

Vends STE + émulateur PC Supercharger (2400 francs) + jeux STE falcon 1 et 2 (300 francs) + Builderland (100 francs). Monsieur Houelle, 48 avenue Préssensé 69200. Tél: 78 09 08 34.

Vends STF double face + SM 124 + Ultimate Ripper + Joystick + nombreux livres + très nombreux softs. Offre TI-95 Procalc en cadeau! Tél: 62 39 71 41, le soir.

Vends 520 STE Atari + moniteur couleur + manette + nombreux jeux + boite de rangement + kit téléchargement, sous garantie, 3700 francs. Tél: 35 37 26 36, après 20 heures.

Vends 520 STF + Péritel + originaux (Batman, etc) + démos + 10 disks vierges, 2000 francs. Tél: 20 49 90 71, région Lille.

Vends Atari 520 STF + free boot + 40 disks + jeux originaux + livres, le tout pour 2200 francs, à débattre. Préférence Bretagne ou Poitou. Cherche Archimedes. Tél: 97 37 79 56. Christophe Planty, 5 rue de Ludwigshafen, 56100 LORIENT.

Vends 520 STF (double face) + nouvelle souris + sélecteur de face + Deluxe, Maupiti, Space Quest III, LC10 couleur + SC1224 couleur + disquettes + boite + jeux, 6490 francs. Tél: 56 74 95 24.

Vends 520 STF 1Mo, moniteur couleur, 2 joysticks, 200 disks, 2 boites, Flight Simu-lator, 9 originaux, cable Null Modem, 3 livres, 75 journaux, 8800 francs à discuter. Nicolas-Romain Blaizot, Domaine du Hautbois, 14800 Deauville. Tél: 31 88 78 38.

Vends Atari 520 STF gonflé à 1Mo (2500 francs), moniteur couleur Philips CM8832-11 sous garantie (1500 francs), lecteur externe 3"1/2 Roctec RF302R sous garantie (500 francs). Tél: 94 83 91 21, après 21 heures.

Vends Atari 520 STF gonflé à 1Mo + moniteur couleur SC1425, 3500 francs le tout. Tél: 27 81 98 93, uniquement Région Nord.

Vends Atari 1040 STF + disque dur 20Mo + AT Speed + moniteur monochrome et adaptateur couleur + nombreux softs PC et Atari, le tout pour 7500 francs (région Caen). Tél: 31 43 92 69.

Vends 1040 STF étendu à 3Mo + SM124 + SC1425 + inverseur vidéo, 5000 francs. Tél: 27 25 92 25, après 20h.

Vends 1040 STF + moniteur couleur CM 8802 + moniteur monochrome SM124 + lecteur externe + Free Boot + Trackball + joystick + livres et revues + nombreux programmes (jeux, programmation, aventure), TBE, 6800 francs. Tél: 94 47 92 09.

Vends 1040 STF Janvier 89 (1900 francs), disque dur Megafile 30 Janvier 90 (2500 francs), le tout en très bon état. Tél: 48 65 58 27, Bourges.

Urgent! Vends Atari 1040 STF (TBE, 06/89) + moniteur couleur SC1425 + freeboot + utilitaires + jeux + démos + boite (Note Du Claviste Feignasse Comme Pas Deux: boite? Eh, les kids, si vous commencez à vendre des boites, des cartons, des emballages, je suis pas prêt de me reposer...) + manette : 4800 francs! Tél: 27 26 67 30 après 18h, Cyrille.

Vends 1040 STF + moniteur monochrome + SH205 + souris + divers, le tout en bon état, 4500 francs à débattre. (Note Du Claviste Effaré: et ton numéro de téléphone, je le tire aux dés? Baaaaanane!)

Vends Atari 1040 STF monochrome + disque dur 30Mo + lecteur 5"1/4 + imprimante Brother 1109 + boitier alimentation 200W + logiciels: 9000 francs. tél: 28 26 87 08 ou 20 47 09 80 (NDC: 80? 50? Ecrivez proprement, svp).

Vends Atari 1040 STF + SM124 + imprimante Star NL10 + accessoires + nombreux logiciels, pour 5000 francs. Tél: 80 92 18 87.

Vends Atari 1040 STF + SM125 + imprimante, 80 logiciels et divers accessoires, sur département 42. Tél: 77 80 16 10, Romuald.

Vends 1040 STF Freebooté, très bon état, très bas prix, avec ou sans nombreux logiciels. Tél: 74 58 60 15, Hervé. Recherche logiciels professionnels.

Vends Atari 1040STE (2Mo) + disque dur Omikron 32Mo + SM124 + drive externe + handy Scanner, 7000 francs. Tél: 42 79 56 02, Christian Roucaimol, Vitrolles.

Vends Mega ST1, moniteur monochrome, disque dur 20Mo, lecteur externe, imprimante 9 aiguilles, accessoires, logiciels, revues, pour 9500 francs. Vente séparée possible. Tél: 65 30 10 96.

Vends Mega ST1, moniteur couleur, disque dur 30Mo, digitaliseur vidéo Vidi ST + Miximage, livres, logiciels, pour 12000 francs. Tél: 41 48 24 64, après 19h, Philippe.

Vends Mega ST1 mono (3500 francs), disque dur Megafile 30 (3000 francs), imprimante 9 aiguilles (800 francs). Tél: 53 54 14 32, Dordogne.

Vends Mega ST1 + disque dur 32Mo + écran couleur + imprimante Star LC10 + très nombreux logiciels + joystick. Le tout pour 10000 francs. tél: 76 90 45 70, après 18h.

Vends Mega ST1, garantie 11 mois, souris, joystick, logiciels programmation, bureautique, jeux, 5000 francs à débattre. Tél: 79 08 62 17, Eric Fournier.

Vends Mega ST1 + écran monochrome + divers: 3700 francs. Tél: 91 81 78 34.

URGENT. Vends Mega ST1 (Juin 89) + moniteur couleur + SM124 + LX800 (Janvier 90) + drive Cumana 3"1/2 DF 720Ko + livres + GfA Basic + 100 disquettes + joystick + tapis souris + switcher de moniteurs + câble Minitel, le tout 6400 francs. Tél: 99 00 10 07.

Vends Mega ST1 + Megafile 30 + moniteur monochrome (le tout d'octobre 90) + Le Rédacteur v3.10 + Timeworks + nombreux logiciels originaux. Tél: 49 67 10 27 le week-end (département 79).

Vends Mega ST1 4Mo + Megafile 30 + SM 124 + piano-synthé Midi + divers, 11000 francs. P. Pavan, 2 rue du Lac, 25660 SAONE. Tél: 81 55 80 77.

Vends Mega ST2 + logiciels (5000 francs), disque dur SH205 (2000 francs). Tél: 98 48 22 13, après 19h, ou le week-end.

Vends Mega ST2 + SM124 + lecteur externe 3"1/2 double face +130 logiciels: Le Rédacteur 3, Calamus, VIP Pro, Fun Face, Le Dessinateur, Outline + musique + langages + utilitaires + jeux + disquettes + boites de rangement + filtre + revues, le tout 9500 francs. Tél: 53 07 02 89, heures repas.

Vends Atari Mega ST2 + SC 1224 (6400 francs), disque dur 85Mo (DMA + SCSI) sous garantie (3600 francs). Tél: 61 08 57 92, ou 61 32 82 50.

Vends Mega ST2 + disque dur 20 Mo + logiciels, peu servi. Tél: 99 41 74 60, en semaine, le soir.

Vends Unité Centrale Mega ST2, excellent état, 3500 francs à débattre. Tél: 88 41 72 35, heures bureau, demandez Jean-Luc Evrard, Strasbourg.

Vends 520 STF, 2.5Mo de RAM + Megafile 30 + drive externe + freeboot + moniteur SM124 + nombreux logiciels, 7500 francs. Tél: 44 87 02 65 (bureau), 23 96 12 22 (le soir).

Vends Atari 520 STE à 2Mo + joystick + souris + 50 disquettes (PAO, MAO, jeux). Faire offre le week-end. Cherche contacts Mega STE. Tél: 73 72 20 75, Sébastien.

Vends Mega ST4 Tos 1.4 + disque dur 30Mo + écran haute résolution + imprimante laser + 2 toners neufs, pour 12500 francs. Dernières versions Calamus, Le Rédacteur, Calligrapher Pro, Calcomat 2+, etc. Tél: 41 87 18 31 (Angers).

Vends ST4 + Megafile 60 + PPM, Calamus, Arabesque, Lazy Paint, etc, pour 10000 francs. Scanner Canon X12, 5000 francs. Tél: 44 78 45 95.

Vends solution complète micro-édition: Mega ST4 + SM 124 + disque dur 30 Mo + Handy Scanner 400 dpi + très nombreux logiciels, le tout pour 16000 francs. Tél: 56 81 91 21.

Vends Atari TT30, 8Mo + écran couleur multi-sync + écran A3 + CD Rom Atari + Devpac2 développeur + Star LC 24-10, le lot pour 36000 francs ou faire offre séparée. Tél: 61 54 75 30.

Vends écran MegaVision 19" N&B pour Mega ST (7000 francs), moniteur couleur Atari (1200 francs), Epson LX80 + ruban (1200 francs). Tél: 98 47 38 23, laisser coordonnées.

---

vends moniteur monochrome Philips (600 francs), 6 barrettes SIP x 256Ko (400 francs), séquenceur Yamaha QX21 (700 francs). Cherche contacts sérieux STE sur Vendée. Tél: 51 51 19 34, Bernard.

Vends Megafile 30, état neuf (1800 francs), Package Emulation Mac: Spectre GCR + ROMs + T-Script + driver imprimante (2900 francs), Package ST Replay (400 francs). Tél: 56 05 12 53 (domicile), 56 57 32 47 (bureau).

Vends disque dur Megafile 30Mo, excellent état, 2800 francs. Tél: 46 84 33 78, le soir, Rochefort.

Vends digitaliseur Vidi ST + Miximage (900 francs), 4 barrettes SIMM 256ko (300 francs la paire), synthétiseur DX7 + Flycase, état neuf, 7000 francs. Tél: 55 93 28 52 (après 18 heures).

Vends lecteur externe Atari ST neuf sous emballage, simple face, 450 francs + nombreux softs. Tél: 68 84 08 78.

Vends lecteur externe DF + Hardcopieur (1000 francs, sous garantie), cartouche Master Sound + cable (200 francs), compilateur GfA 3.03 (150 francs). Tél: 59 92 16 51.

Vends lecteur de disquettes 5"1/4 360/720Ko + 140 disquettes (1000 francs), moniteur monochrome + adaptateur Péritel (400 francs), 512Ko de RAM (400 francs). Tél: 63 47 58 64, Christophe.

Cause changement scanner, vends Handy Partner (pack: scanner Golden Image + logiciel Image Partner), sous garantie, le tout pour seulement 1500 francs. Tél: 79 25 40 69, Didier.

Lille: vends imprimante Mannessman-Tally 80S (compatible Epson), 9 aiguilles, 700 francs. Tél: 20 23 57 98, après 18 heures.

Vends imprimante Star LC10 très peu servi avec cordon + Calligrapher Junior. Prix: 1400 francs, vente séparée possible. Tél: 99 30 08 61.

Vends imprimante Citizen Swift 24 noir et couleur sous garantie + un ruban noir + 4 rubans couleur + Flexidump+, le tout pour 3300 francs. Tél: 41 87 18 31 (Angers).

Vends imprimante à jet d'encre Diconix 150 Plus (1900 francs), moniteur Commodore 1084 (1300 francs), original Calligrapher Pro (400 francs). Olivier Fabrègue, 18, rue Villons les Buissons, 14000 CAEN.

Vends imprimante couleur Canon PJ1080A. Tél: 68 26 30 71. Robert Vigean, 11380 MAS CABARDES.

Vends SLM804 + DME, peu servie. Tél: 56 04 17 07, heures repas.

Vends Laser SLM 804: 7500 francs. Matricielle 9 aiguilles pour Commodore 64: 700 francs. Tél: 87 30 56 28, Pierre, après 18 heures.

Vends extension mémoire STE 512Ko (350 francs), cartouche Multiface (350 francs), Livre du GfA Basic (100 francs) + 20 disks jeux originaux. Tél: 93 13 49 84 (Nice).

Vends lots de 16 RAMs 41256 pour étendre STF à 1Mo de mémoire, pour 400 francs. Tél: 61 70 46 17, après 18h, Franck.

Brade nombreux livres ST (GfA, C, Assembleur) + nombreux accessoires + docs + softs. Tél: 43 06 65 81.

Vends ou échange: Basic GfA 3.0 (350 francs), livre Trucs & Astuces en GfA (150 francs), Livre du GfA (120 francs), livre 1ST Word+ (150 francs). Cherche Basic 1000D, Transfile ST (pour Sharp). Tél: 96 74 68 67, Gérard, St Brieuc.

Vends cartouche Ultimate Ripper neuve (550 francs). Tél: 61 31 92 54, J-Christophe.

Vends cartouche Multiface (400 francs) Psion Organiser 2 XP + illisible + Datapack 64Ko (1800 francs), cartouche FM Melody Maker pour STF (500 francs). Tél: 88 95 07 87.

Vends originaux Tennis Cup et Lemmings 300 francs chacun. Frédéric, 47 rue Lamendin, 62138 DOUVRIN. Tél: 21 40 90 02. (Note du Claviste Las: eh bé et ton nom, fada?)

Vends GfA-Assembleur/ Debugger (300 francs), GfA 3.5 (100 francs), Multiface ST (350 francs), Cyberpaint, le tout sur Atari ST. Tél: 89 66 45 50.

Vends démos ST les plus récentes, enveloppe timbrée pour liste complète. David Ducassou, route de Monségur, 40700 HAGETMAU. Tél: 58 79 43 01. Qui peut m'imprimer des PI1 en couleurs?

Urgent! Vends originaux, livres sur STF/E, vends lots disks 3"1/2 (par 10), boites posso. Sylvain Bail, 5 avenue Louis Blériot, 44340 BOUGUENAIS.

Vends pour Atari STF/E: Dames Simulator (140 francs) + Beach Volley (150 francs). Thierry Durand, 5 allée Paul Gauguin, 47510 FOULAYRONNES. Tél: 53 95 62 26, le soir.

Vends originaux pour ST: ST Replay Pro (1000 francs), GfA-Raytrace (350 francs), GfA Artist (350 francs). Tél: 94 95 40 82, après 17 heures, Alexandre.

Vends Pack GfA (disks + manuels), neuf, 700 francs le lot, ou Basic 3.07 + compilateur 3.03 (500 francs), GfA Assembleur (300 francs), Kit téléchargement (80 francs). Tél: 56 37 24 56, le soir, Patrick, à Pessac.

Suite vente ST, reste 15 originaux, revues et démos. Stéphane Thullier, 9 rue G.Basquin, 59810 LESQUIN.

Vends démos et disks magazines les plus récents. Liste contre un timbre (Note Du Claviste Etonné: c'est quoi ce trafic?), sérieux et rapide. David Ducassou, route de Monségur, 40700 HAGETMAU.

Vends ou échange: GfA 3.0 (350 francs), Livre du GfA (120 francs), Trucs et Astuces en GfA (160 francs), Livre de 1ST Word (180 francs). Cherche Basic 1000D, Transfile ST. Tél: 96 74 68 67.

URGENT. Vends Super Charger 1Mo version 1.50 avec nombreux softs PC (Windows, PC Tools, etc), neuf, pour 2500 francs. Vends IMG Scan (800 francs), lecteur externe 5"25 40/80 pistes (1000 francs). Tél: 72 51 17 94, P. Mathé, le soir.

Pour imprimante 9 aiguilles: soft de téléchargement de polices qualité courrier, 60 francs tout compris (Note Du Claviste Pas Chié: voir aussi Utils_FX80 dans ce même canard, pas cher, beau). Tél: 99 99 31 98, en soirée.

Vends pour 520 STE jeux originaux: Explora 3 (150 francs), Powermonger (150 francs), Sim City (150 francs), BAT (200 francs). Tél: 86 65 37 08, Philippe.

Vends originaux + boites: RVF Honda (100 francs), Austerlitz/ Waterloo (90 francs), Midwinter (150 francs), F-19 (150 francs), Resolution 101 (120 francs), Sherman M4 (100 francs), ST/STE. Tél: 78 84 88 96.

Vends très bas prix ou échange plus de 500 démos et Freeware pour ST/STE (musiques, pirates, X, utilitaires). Demandez ma liste gratuite à: Frédéric Houet, 20 rue des Moissons, 51100 REIMS.

Vends pour ST/STE STOS Basic 2.5 (120 francs) + programmes + STOS Paint, Colossus Chess X (120 francs), le tout d'origine. Tél: 78 84 88 96, Stéphane.

Vends logiciels DP et originaux jeux, à partir de 5 francs (démos, jeux, utilitaires...). Demandez ma liste. Tél: 22 51 84 01 ou 22 51 96 85 après 18 heures.

Vds logiciels LE REDACTEUR 3.01 (500 francs) et GESTOCK 3.03 (600 francs). Tél: 32 31 23 36 (heures bureau).

Vends jeux originaux sur ST (entre 50 et 100 francs): Operation Stealth, Titan, Skweek, Barbarian 2, Knight Force, Bloodwych, Tiger Road, Krypton Egg, GfA 2.02, Fun Face, Maths CM (Note Du Charmant Claviste: ce sont des jeux, ça?). Téléphonez vite au 93 74 41 41. Matthieu Mangan, La Pinède de Valbosquet, 905 chemin du Malbosquet, Villa 5, 06600 ANTIBES.

Vends Supercharger (2300 francs), Mega ST1 (3900 francs), disque dur SCSI 42 Mo (2500 francs), Notator 3.0 (3300 francs), serveur Must (1200 francs). Tél: 78 25 12 20.

Vends émulateur PC Supercharger 1Mo très peu servi. Tél: 59 30 50 69, Pau.

Vends Spectre GCR 2.65C, garanti jusqu'à fin 91. Tél: 73 27 91 48.

Vends Téléstrat Oric (Note Du Claviste Donnant Dans Le Private Joke: une victime de plus!) avec 2 drives 3" + K7 de jeux + livres + Théoric. Tél: 96 47 04 54, le soir, Yann.

Vends Oric Atmos 48Ko (Note Du Claviste Plié: mais c'est un complot!) + lecteur disquette Jasmin 2 + bloc alimentation + logiciels utilitaires et jeux + livres. Imprimante Seikosha GP500A, parfait état. Revues "Théoric" 1 à 37. Faire offre. Tél: 27 49 94 52.

Vends Tandy 1000EX, moniteur couleur, imprimante Tandy DMP132 + docs, jeux, divers, le tout pour 6000 francs. Tél 66 53 17 58 de 15 à 18h.

Vends Portfolio + interface parallèle + cables: 1200 francs. Vends imprimante DMP compatible Epson, 500 francs. Tél: 21 84 07 37, après 18h.

Vends Olivetti portable 7 kilogrammes M211 43V, année 90, 40Mo, 16000 francs (valeur 38000), très bon état. Tél: 40 64 82 01.

Vends Amstrad 464 + moniteur couleur avec le guide Amstrad ainsi que de nombreux jeux, le tout en très bon état: 2000 francs. John Maes, 2 rue du Parc, 59100 ROUBAIX. Tél: 20 24 79 56.

Vends C64 + lecteur disks + imprimante Star SG10C (1000 francs seule) + nombreux jeux et utilitaires, 2500 francs. Tél: 56 85 36 12, Bordeaux.

Vends C64 + unité K7 + unité disquettes 5"1/4 simple face + imprimante MPS801 + adaptateur Péritel + Synthétiseur vocal + logiciels + livres, possibilité moniteur, prix 2000 francs. Tél: 99 55 88 50.

Vends kit Z80A microprocesseur MPF 1B, alimentation + docs, 1500 francs. Tél: 87 82 32 98, Albert, après 18h.

Vends synthé Casio CT670, 2500 francs à débattre. Vends CD Port Philips, 600 francs (valeur 1200 francs). Cherche Synthé K1 pas cher. Tél: 53 04 27 21.

Vends Track 24 (300 francs), Studio Conceptor (500 francs), Sim City (200 francs). Tél: 67 26 92 16, entre 15h et 20h.

Vends Pro24 v3.0 + clé + doc, 1500 francs. Tél: 43 03 20 42, Mayenne, Jean-Michel, week-end.

ACHATS PROVINCE

Achète moniteur monochrome SM124 d'occasion. Faire proposition à Monsieur Commageat Francis, 73 avenue du Capitaine André Buffet, 19100 BRIVE.

Achète Great Courts 2 et Strip Poker 2+ uniquement sur 1040 STE (à prix budget)! Laurent Meslin, 86 rue Thiers, 33500 LIBOURNE. Tél: 53 71 17 65.

Recherche digitaliseur vidéo "Vidi ST" pour Atari. Faire offre au 61 64 99 23, Eric Lamant.

DIVERS PROVINCE

Cherche dans la région de Rouen un club dédié au ST. Jérôme Balant, 11 rue Léon Devaux, 76240 BONSECOURS.

Réalise interface SCSI avec logiciels pour disque dur, scanner, réseau, etc. Pour ST, Mac, PC. Tél: 19 32 41 62 51 36.

Echange jeux Atari contre jeux intéressants. Pirateurs s'abstenir. Tél: 19 3 259 510 783, après 19 heures.

Vends programme numérologie en GfA 3, création personnelle pour monochrome. Programme: 50 francs, programme + source: 70 francs. Monsieur Blassiaux, chemin d'Aubaygues, 34700 Soubes. Tél: 67 44 35 74. Peux le convertir en GfA 2.

Lyon: échange collection de timbres contre chassis, volant, roues, ou moteur d'Alfa Sud. Contactez OYA SHIVO sur le 3615 JELAIDANLOS.

Recherche graphistes, musiciens, coders 68000 sur STE pour créer des démos. Envoyez disk à Thierry Durand, 5 allée Paul Gauguin, 47510 FOULAYRONNES.

Cherche renseignements sur le logiciel "Anamorphoses à miroir" (ST Mag n°11). Tél: 91 81 78 34.

Etudiant donne cours programmation en GfA 3: 30 francs l'heure. Tél: 20 02 71 69, Mickael.

Cherche dans région Lyonnaise contact sérieux pour musique, possède PRO 24 III, U-220, 21 ans (possibilité création groupe (Note Du Claviste Hétéro: et plus si affinités?). Tél: 78 84 88 96, Stéphane.

Codeur recherche graphistes et musicien(s) dans le 29 pour création de démos... Tél: 98 07 88 67, Patrice.

URGENTISSIME! Recherchons coders, graphistes et musiciens pour création démos ou autre sur Angers. Tél: 41 44 44 81 (David).

Lyon. Cherche nouveaux membres pour l'association "Les Amis de Klaus", réunions, débats, promenades, contactez OYA SHIVO sur 3615 PLUCEGRANPLUCEBETE.

Petites Annonces

Cherche tête imprimante Citizen Swift 24 (même en panne), échange contre livres ou logiciels originaux. Tél: 68 26 30 71. Robert Vigean, 11380 MAS CABARDES.

Cherche programmes Midi et Midifiles, et vends synthé Casio MT260 Midi pour Atari, 1500 francs. P Pavan, 2 rue du Lac, 25660 SAONE.

REGION PARISIENNE VENTES

Vends Atari 520 STF + joystick + prise Péritel + souris + jeux (news), pour 3200 francs à débattre. Tél: 60 60 63 24, Guillaume.

Vends Atari 520 STF étendu à 1Mo, Tos 1.4 + moniteur SM 125 + disque dur SH205 + drive externe 720 Ko + joystick + utilitaires (GfA 3.0, dessin, traitement de texte, tableur, etc) + jeux. Le tout pour 5000 francs. Tél: 47 86 13 12, le soir, Fabrice Lacharme.

Vends 520 STF simple face, Calcomat+, GfA 2, Datamat, Flight Simulator, Gunship, Starglider, Crazy Cars, 1500 francs. Tél: 48 44 88 52, James.

Vends Atari 520 STF + moniteur couleur SC1425 + Citizen 120D + Manoir de Mortevielle, ZZ Rough, Mac3 + nombreux jeux et utilitaires, le tout 5000 francs, TBE. Tél: 46 68 07 09, Laurence (Note Du Joli Claviste: purée, une nana qui vend son ST, c'est d'un triste!).

Urgent! Vends 520 STE exceptionnel, super moniteur couleur SC1435 + Cyber Studio + jeu. Le tout cédé pour 4600 francs. 91230 MONTGERON. Tél: 69 03 52 33, Frédéric.

Vends 520 STF + moniteur couleur + revues + logiciels, 3500 francs. Tél: 48 44 10 37.

Vends 520 STE, peu servi + joystick + 3 jeux originaux, 2500 francs. Tél: 60 10 40 15, Jean-Baptiste, week-end seulement.

Vends Atari STF + joystick + souris + jeux originaux + livre + revues, 3000 francs. Vends lecteur externe double face 3"1/2, 500 francs. Tél: 39 57 64 89, Azdine.

Vends Atari 520 STE + moniteur couleur SC 1224 + 50 jeux et utilitaires + émulateur PC + souris + joystick + manuels, un an, 4200 francs à débattre. Tél: 34 16 05 25, Guillaume, Val d'Oise.

Vends 520 STE 1Mo, lecteur 3"1/2, SM 124, SC 1425, switch, joystick, nombreux softs (jeux, Script II, Paint Designer, Degas, etc), valeur 12000 francs, vendu 7000 francs. Tél: 45 31 23 59 après 20h.

Vends Atari 520 STF, double face, très bon état, extension mémoire 1Mo + free boot installé + nombreux magazines (Joystick, ST Mag, Gen4) + nombreuses docs (GfA Basic, Neochrome, Logo...) + nombreux disks + souris + joystick + housse + cables sous emballage, pour 3600 francs. Tél: 60 84 08 98.

Vends 1040 STF (3500 francs), imprimante (1000 francs), MT32 (1500 francs), Spectre GCR (2500 francs), le tout 7000 francs. Tél: 69 00 91 34, P. Mahé.

Vends 1040 STF + SM 124 + Megafile 30 + Star LC 24-10 + nombreux logiciels + meuble informatique. Jamais servi! 10000 francs. Tél: 45 76 88 04, après 19h, Laurent.

Vends 1040 STE + souris + cable Péritel + jeux (BAT, Crazy Cars 2, etc) + utilitaires + GfA Basic + Devpac, très bon état, pour 3500 francs. Tél: 43 76 68 42, après 18 heures.

Vends 1040 STF + moniteur SM124 + lecteur externe 3"1/2 + imprimante Star LC10 + Spectre GCR 3.0 + logiciels ST et MAC: 6000 francs. Tél: 46 83 69 90 ou (16) 44 57 18 39.

Vends Atari 1040 STF + moniteur couleur + imprimante Star LC 10 2ème édition + programmes, etc, 6000 francs à débattre. Tél: 45 58 09 20, Julien J-M.

Vends Atari 1040 STF + moniteur monochrome SM124 + imprimante Citizen 120D + joystick + cable Péritel + softs + livres. Le tout en très bon état: 3900 francs. Tél: 69 07 14 19, Monsieur Hello, en soirée.

Vends Atari 1040 STF + moniteur couleur Commodore 1084 + lecteur externe + imprimante Star LC10 + cable Minitel + programmes professionnels + jeux, le tout pour 4800 francs. Tél: 45 00 98 79, Christian.

Vends 1040 STF + carte PC-Ditto 2 installée sous garantie + carte horloge: 3500 francs, ou échange contre Mega ST. Tél: 39 68 68 73, le soir, Patrick Greslin.

Vends Atari 1040 STF + moniteur monochrome SM125 + imprimante matricielle Fuji + lecteur externe Cumana + disque dur 20Mo + nombreux logiciels (Rédacteur...), 7000 francs. Tél: 34 77 21 57.

Vends 1040 STF + moniteur couleur SC1425 + 2ème lecteur SF314 Atari + souris + joystick + logiciels: 4800 francs. Tél: 45 69 51 38.

Vends Mega ST1 + moniteur couleur SC1425, parfait état, logiciels + souris + joystick, le tout pour 6000 francs. Tél: 60 68 45 92, Melun.

Vds Mega ST1 + Megafile 30 + moniteur SM 124 + nombreux logiciels (utilitaires, jeux). Tél: 31 26 84 58.

Vends Mega ST1 + SM124 + disque dur Megafile 30 + lecteur 5"1/4 + programmes & livres (sous garantie), 8000 francs. Tél: 47 84 18 49, Bruno.

Vends Atari Mega ST1 + moniteur monochrome SM 124 + switcher écran, 4000 francs. Gilles Bourdon, 25 boulevard d'Igny, 91430 IGNY. Tél: 69 41 38 15.

Vends Mega ST1 + SC1425 + disque dur 30Mo + 17 jeux + 21 utilitaires + n°1 à 21 Atari Mag + 16 ST Mag, 14000 francs à débattre. Tél: 47 48 90 11, Stéphane après 20h30.

Vends Mega ST1 + moniteur couleur SC 1224 + Megapage + Supercharger V1.4 + DOS 4.01 + logiciels PC + logiciels ST (Tdt, PAO, dessin, jeux...), 7500 francs. Tél: 48 60 13 30, après 20 heures.

Vends Mega ST1 (garanti un an) + moniteur couleur + moniteur monochrome + imprimante matricielle 9 aiguilles + logiciels et commutateur vidéo, 7100 francs. Tél: 37 83 59 78.

Vends Mega ST1 nu, très peu servi, prix à débattre. Tél: 85 80 91 15.

Vends Mega ST1 + moniteur SM 124 + logiciels Calligrapher Pro + LDW Power + Wordup 2.1, l'ensemble ou séparé. Tél: 47 90 88 47.

Vends 520 STE 2Mo + écran + imprimante, 8000 francs, le tout sous garantie + disquettes jeux, PAO, musique, tous originaux, liste sur demande. Tél: 48 59 73 49, après 21h.

Vends 1040 STF 2,5Mo + PC Ditto 2 (3800 francs), LQ500 Epson (2000 francs), Handy Scanner (1000 francs), Spectre GCR (2500 francs). Tél: 48 40 64 91.

Vends Mega ST2, parfait état + moniteur SM 124 + SC 1224 + lecteur 5"1/4 + 320 disques. Le tout vendu 8000 francs (valeur 14000). Tél: 34 89 45 54, Alex, après 18 heures, Yvelines.

Vends Mega ST2 + moniteur SM125 + modem CAP23 Minitel + souris optique, pour 5000 francs. Tél: 49 80 21 00, Monsieur Denis.

Vends Atari STE 2Mo (2900 francs), disque dur Megafile 30 (2600 francs), écran couleur Commodore 1084S (1700 francs), écran monochrome Atari SM125 (900 francs). Tél: 43 03 42 37, Noisy le Grand.

Spécial programmeur: vends STE 2Mo + Megafile 30 + SM 124 + ST Replay 8, 8000 francs, livré avec C, GenST2, GfA, Omikron, Ressource Editor et plus de 20Mo de sources. Tél: 43 25 45 49.

Vends Mega ST2 gonflé à 4Mo + SM 124 + disque dur Megafile 30, 7000 francs, garanti Décembre 91. Vends nombreux logiciels dont Arabesque Pro (1000 francs), Timeworks (500 francs), G+ Plus (200 francs). Tél: 45 81 45 31, Patrick.

Vends STE 4Mo (4200 francs), Megafile 30 (2800 francs), barettes STE 256Ko (130 francs), 1Mo (320 francs). Tél: 48 49 86 41, Jacky.

Vends Atari Mega ST1 à 4Mo avec disque dur 30Mo, très bon état, garanti 18 mois, le tout à 8900 francs sans écran (livré avec anti-virus, répertoire vidéo, répertoire adresses, Sapristi, etc...). Tél: 45 84 01 22.

Vends Atari ST4 + Profiler 44 + SLM 804 + SM124 + camera N/B + Spectre GCR + nombreux logiciels, 25000 francs. Tél: 43 25 64 64, le soir.

Vends Mega ST4 + SM 124 + souris neuve + manette + logiciels PAO (Calamus, Outline, Didot), le tout pour 6900 francs. Tél: 48 56 17 77.

Vends Mega ST4, bi-TOS, monochrome + couleur (Thomson), clavier RTS Mouse Genius 7, lecteur externe, 6500 francs. Tél: 49 63 02 73, Monsieur Langlois.

Vends Mega ST4 + SM124 + SC1425 + AT Speed, en très bon état, emballage d'origine, nombreux logiciels et jeux: 12000 francs. Tél: 48 93 07 35, après 18h, Pierre.

Vends Mega ST4 + écran monochrome + écran couleur + switcher + joystick, 9000 francs. Vends Lynx + 6 cartouches: 2300 francs. 1040 STF (alim à réparer): 2000 francs. Tél: 47 36 08 28, Eric.

Vends station PAO complète: Mega ST4 monochrome + Megafile 30 + laser SLM 804 + scanner type 10 + Publishing Partner Master + Image Partner + Reading Partner, le tout indissociable. Tél: 46 57 50 15, le soir, Pierre.

Vends 520 STE 4Mo RAM (5000 francs), moniteur couleur SC 1224 (1800 francs), ou l'ensemble pour 6500 francs. Tél: 60 60 09 79.

Vends extension 1Mo pour 520 STE: 250 francs. Disquettes vierges TDK DFDD: 50 francs les 10. Tél: 39 59 60 88, Philippe.

Vends 2 mémoires SIMM 256Ko pour PC, Atari, Amiga, Mac: 180 francs. Vends disque dur 10Mo pleine hauteur, 450 francs. Tél: 48 39 34 50.

Vends barrettes SIP 256Ko, 300 francs les deux. Tél: 47 80 61 05, Pascal, répondeur.

Vends boitier Tower pour Mega ST + alim + connectique, 1000 francs à débattre. vends interface DMA-SCSI, 600 francs sous garantie. Tél: 69 43 38 79, Alexandre, après 20h.

Vends Laser SLM 804 garantie un an avec toner, cédée 6000 francs. Tél: 16 37 23 00 06 (département 28, à la limite du 78).

Vends imprimante Star NL 10, compatible Epson FX80. Installe extension RAM 2Mo sur 520 STF, Protechnic. Tél: 47 41 56 51.

Vends imprimante Image Writer (1500 francs), lecteur 5"1/4 (600 francs). Tél: 30 90 78 49, Alain ou Nathalie.

Vends scanner Canon IX12F A4 à plat (5000 francs), laser SLM 804 2000 sorties (5300 francs), Laser Postscript Turbo AST (17000 francs). Tél: 69 30 91 46, Eric.

Vends scanner à main Logitech avec repro Studio Jr, 32 niveaux de gris, 400 dpi (Pack ALM) pour 1500 francs, excellent état. Tél: 64 97 96 11, Arnaud.

A saisir, moniteur monochrome Atari SM 124 (neuf, allumé deux fois) (Note Du Claviste Con: et éteint au bout de combien de temps? On nous cache quelque chose!), 700 francs. Vends aussi commutateur monochrome/couleur, 50 francs. Tél: 30 65 24 54, avant 17 heures, Monsieur Aouidad.

Vends moniteur couleur SC 1425: 1850 francs. Tél: 43 06 38 63.

Vends moniteur couleur Philips + cordon Péritel pour ST (RGB CM8535), 650 francs. Vends convertisseur Midi Roland CP-40, 1300 francs. Vends Steinberg Pro12, 200 francs. Tél: 48 70 03 30.

Vends moniteur couleur Europhon (Pal/Secam), 1300 francs. Tél: 64 45 71 24, laisser message.

Vends moniteur couleur SC1435 (1400 francs), Mark Williams C 3.09 + debugger source (700 francs. Neuf: 2000 francs). Tél: 42 53 84 71, Thierry, répondeur.

Vends disque dur Seagate ST 225 20Mo + interface DMA SCSI sans carte adapteur, 1100 francs. Tél: 60 01 58 06.

Vends disque dur SH205 (2000 francs), Le Livre de Gem, le Langage Machine sur ST, Musique et Midi, Graphismes et Sons sur Atari ST (300 francs). Tél: 60 13 06 67.

Vends disque dur SH205 (2000 francs), carte DMA-SCSI converter (500 francs). Tél: 30 76 07 72, le soir après 20 heures.

Vends Megafile 30 + Time Works, très bon état, 3000 francs. Lecteur externe 5"1/4 (40/80): 1000 francs. Supercharger 1Mo v1.4 TBE: 1900 francs. Tél: 39 80 77 37, Olivier, après 18 heures.

Vends magazines: Tilt (du 57 au 69) et ST Mag (34-49), 10 francs le numéro. Possibilité contact ST. Tél: 46 83 19 43.

Vends livres: GfA Gem Up, Le livre du GEM, Applications sous Superbase, moitié prix. Tél: 48 51 53 37, le soir.

Vends livres: GfA Basic (100 francs), GfA 3.0 (100 francs), GEM (100 francs), langage machine (70 francs), graphisme (70 francs), graphisme 3D (100 francs). Tél: 43 75 92 66, poste 242, J-L Vannier.

Vends jeux originaux, 100 francs l'un + GfA Basic 3.07 + compilateur 3.03 + GfA Assembleur (450 francs) + Ultimate Ripper (400 francs). Tél: 46 28 62 92, après 18h.

Je vends, cause départ, 4 logiciels originaux, neufs et complets, pour Apple Macintosh, à la moitié de leur prix public: Adobe Illustrator (2950 francs), Quark XPress (3750 francs), Aldus PageMaker (4050 francs), Aldus FreeHand (2975 francs). Tél: 47 83 58 89.

Vends Spectre 128 sans les ROMs (1200 francs), PC Ditto II (1700 francs), programmes gratuits avec chaque émulateur. Cherche contacts en émulation. Tél: 43 48 37 70, Paris.

Vends jeux originaux STF/E avec doc et coffret, état neuf: Robocop 2, The Spy Who Love Me, Cadaver, de 18 à 20 heures au 64 06 72 88, ou ruelle du Puits Pinart, 77111 SOLERS, demander Christophe.

Vends originaux Atari avec factures: Publishing Partner Master 1.8 + cinq disques images: 1000 francs; 40 softs originaux à 25% du prix original. Tél: 48 79 06 58, Daniel.

Vends assembleur Devpac ST2 neuf (jamais servi) + modem: 800 francs. Tél: 48 22 73 19.

Urgent! Vends Ultimate Ripper + docs, 810 francs. Sylvain Pillard, 37 rue de l'Avenir, 93240 STAINS. Tél: 48 26 11 47.

Vends originaux: PC Speed STE (990 francs), House Music (200 francs), ST Replay (300 francs), FM Melody Maker (400 francs), etc. Vends 1040 STFC, prix à débattre. Tél: 43 58 60 37, demandez Jipé.

Vends originaux sur ST: Powermonger, Double Dragon, Starray, Thunderblade, 100 francs pièce. Nicolas Bronard, impasse Jean jaurès, 95540 MERY SUR OISE. Tél: 34 64 84 06.

Vends Devpac ST 2 (français, original, neuf) (voir ST Mag 49), valeur 875 francs, vendu 490 port compris. Tél: 42 36 48 70. Cyrille Richard, 50 rue Etienne Marcel, 75008 PARIS.

Vends nombreux jeux originaux sur Atari 520 ST à moitié prix ainsi que diverses revues informatiques. Urgent. Tél: 45 84 26 54.

Vends nombreux originaux, 100 francs l'unité, 200 francs les 3: Xenon 2, Forgotten Worlds, Hard Drivin', F16 Combat Pilo, Dungeon Master, Starglider II, etc. Tél: 34 77 98 05, Yves, après 20h30.

Vends ou échange originaux pour Atari: Operation Stealth, Voyageurs du Temps, Pack GfA 30, Gescomptes - Gesbord, Deluxe Paint ST. Recherche simulations, Ultimate Ripper, Music Master sans MV16. Tél: 64 29 11 59, Christophe, après 19h.

Vends tableur graphique LDW Power, 300 francs. Tél: 64 80 72 96.

Vends Supercharger 1Mo v1.40, état neuf, très peu servi: 1900 francs. Tél: 34 62 39 36, Gilles.

A vendre logiciel "Protos", disquette et doc d'origine, 160 francs. Tél: 43 34 82 07, heures bureau.

Vends logiciels originaux Timeworks, Arabesque, Le Rédacteur, le tout 1000 francs. Tél: 34 86 05 22, le soir.

Vends lunettes Stéréotek pour CAD3D, permettant d'obtenir une vision 3D stéréostopique saisissante: 1000 francs. Tél: 34 62 39 36, Gilles.

Vends PC-AT 386-25 + disque dur 44Mo + 5Mo RAM + lecteur 5"1/4 1,2Mo et lecteur 3"1/2 1,44Mo + écran multi-sync Sony (1024x768) + carte graphique 1024x768 256 couleurs + souris + Windows 3, etc, pour 15000 francs, encore sous garantie, acheté le 16/12/90. Tél: 43 28 09 44, département 93-94.

Vends compatible 256-16, 2 lecteurs de haute densité, disque dur 40Mo, écran monochrome, 2 ports série, un parallèle, pour 8000 francs. Matériel neuf, un mois d'utilisation. Tél: 48 79 16 27, Simon, après 22 heures.

Vends console portable Atari Lynx + California Games + 2 jeux, très bon état, le tout 1000 francs. Contactez Christophe Torki, 4 rue Louise Lesieur, 94700 MAISONS ALFORT. Tél: 43 96 01 62, après 19h.

Vends Sega (1 an) avec 5 jeux + 3 jeux intégrés + 2 manettes + pistolet, 2000 francs. Tél: 64 04 06 89, Alexandre.

Vends Amstrad PCW9512 complet avec imprimante, moniteur noir, disquettes, 4300 francs. Tél: 43 06 38 63.

Vends Atari XE + jeux très bon état, valeur 2500 francs, vendu 1800 francs. Tél: 43 82 54 56, après 18 heures, Thierry. 94190 VILLENEUVE SAINT GEORGES.

Vends Mac Plus, lecteur externe, sous garantie, nombreux logiciels, 5500 francs. Tél: 43 28 58 91.

Vends cartes mémoires 64 Ko pour Atari Portfolio, neuves, 600 francs pièce. Tél: 42 08 05 30, après 18 heures.

Vous possédez une HP48-SX et un micro à sortie RS232 (Votre ST!)? Je vous vends le cable de liaison permettant de transférer toutes données entre vos deux machines. Prix: 150 francs (port inclus). Envoyez un chèque à Monsieur Archour, 46 rue Victor Hugo, 94700 MAISONS-ALFORT.

CHOIX *PA SUR 3615 STMAG

Vends Prophet 2000 + sons (5900 francs), D110 + Edit ST + 2000 sons (3600 francs), M20 + câbles (1500 francs). Le tout pour 9800 francs. Tél: 64 36 17 84.

Vends synthé Casio CZ 230S. Prise Midi, 99 sounds (4 programmables), boite à rythmes, etc. + transformateur Casio pour alimentation + doc, 1300 francs. Urgen, très bon état. Tél: 42 05 09 89, Christophe.

Vends synthé programmable à 6 voix Roland Juno 2, 128 sons + Programmeur Roland + Pro 12, très bon état, 4200 francs (valeur 9000). Tél: 43 39 64 06, Olivier Garcia, 94.

Vends Expander TX81Z Yamaha (1300 francs), moniteur couleur Thomson (1500 francs), moniteur monochrome SM 125 (700 francs). Tél: 47 93 03 47, C.S Lévine.

Vends Notator Alpha, 1290 francs. Tél: 47 49 05 00, Eric, heures bureau.

R.P. ACHATS

1040 STF: cherche Maupiti Island, Dragon's Lair 2, Wrath Of Demon, Shadow Of The Beast (Note du Beau Claviste: pouah sur ST!). Stéphane Gouffault, 13 square Beethoven, 78330 FONTENAY LE FLEURY. Tél: 30 45 15 84.

R.P. DIVERS

Graphiste cherche équipe de programmeurs pour création de jeux ou utilitaires. Tél: 48 22 73 19.

Propose une méthode de cours d'assembleur pour débutants sur ST, par correspondance: méthode didactique, sérieuse et efficace. Contactez Cédric Javault (Note Du Claviste Gâteux: tiens ça faisait longtemps!), 82 rue de l'Université, 75007 PARIS. Tél: 45 44 21 05 après 20h.

Enfin un véritable catalogue de domaine public sur ST (Note Du Claviste Au Courant: et DP Mag, c'est du pipi de chat, peut-être?). Recevez notre disquette démo contenant notre catalogue en envoyant un disk 3"1/2 et enveloppe et autotimbrée. Gamme NBC, 43 rue Berzélius, 75017 PARIS. (NDCAC: voilà, alors ça, ça s'appelle de la pub pour une société commerciale, et je veux pas vous dire ce que vous avez à faire, mais à votre place, les boites qui se font connaître par des PAs, je me méfierais...)

EMPLOIS

Graphiste confirmé recherche programmeur confirmé pour créer jeu de rôle, aventure, action. Déjà fait planches, sprites, animation personnages. Tél: 90 50 14 06, Jean-Luc. Très sérieux.

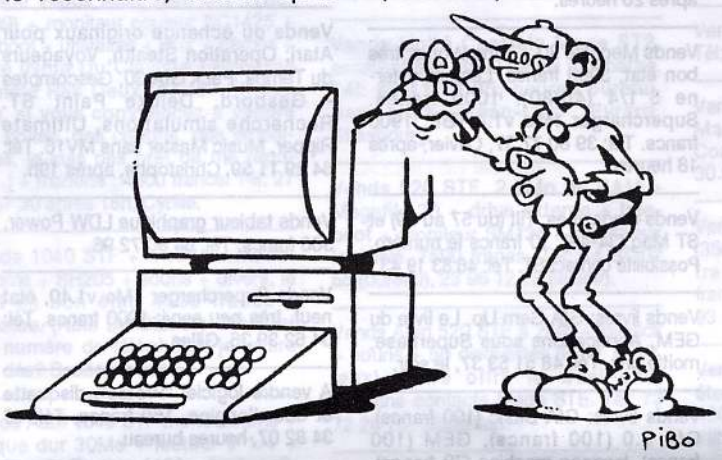
HOMMAGE AUX BONS

Et qui a dessiné les mignons petits crobards tout partout sur toutes les pages? C'est Pibo! Il dessine bien, hein? Purée, je dessinerais dix fois moins bien que lui, je serais content. Cette capacité à pouvoir coucher sur papier ce qu'il y a dans la tête, ce n'est pas donné à tout le monde, avouez que vous en crevez de jalousie. Moi aussi. Pour tout vous dire, j'aimerais pouvoir lui faire manger ses crayons, à Pibo, tellement je l'envie: pourquoi lui, pourquoi pas moi?

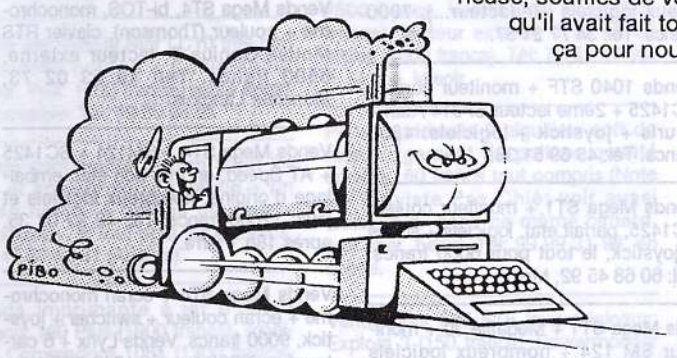
Il y a tout simplement que des bonnes fées se sont penchées sur son berceau, quand il est né, il y a 27 ans. L'une lui a donné le talent, l'autre lui a offert le sens de l'esthétique, une troisième lui a refilé le sens de l'humour et la dernière, qui ne pensait qu'à partir en week-end, lui a donné le pouvoir de résister aux piqûres d'orties, ce qui ne sert quasiment jamais dans la vie d'un illustrateur, mais bon, c'est toujours bon à prendre et Pibo a gardé, à sa place vous auriez fait pareil.

Les lecteurs, ils aiment bien savoir à quoi ressemblent ceux qui leur vendent la camelote. Alors, comme les lecteurs ils y sont allés de leurs vingt francs, on va voir comment c'est fait, un Pibo. Pibo est grand, il est très grand, il est même encore plus grand que ça, c'est utile, je pense, pour un illustrateur ("ha il ne te plait pas mon dessin? -Boum-"), encore que je connaisse un autre illustrateur qui est tout petit, à côté, et qui se débrouillera pas mal le jour où il gommera tous ses vilains côtés, mais je me perds. Bon, Pibo, pour tout arranger, comme s'il n'était pas assez impressionnant comme ça, il faut aussi qu'il soit Rocker, c'est d'ailleurs dans ce circuit Rock qu'on l'a déniché, il dessine pour des magazines, pour des salles et des radios Rock. Voilà pour la description, et les fans ont un auto-portrait pour le reconnaître, histoire qu'ils aillent pas se faire casser la gueule en accostant tous les Rockers balaises qui trainent dans nos rues.

Pour quelqu'un qui ne s'est jamais dépatouillé avec un ST, Pibo s'en sort rudement bien, il n'a pour ainsi dire aucune notion d'informatique, il n'a pas d'ordinateur, pas de Minitel, rien, c'est un Rocker. Quand je lui explique un gag ou une illustration à faire, je le regarde droit dans les yeux pour essayer de faire passer mon idée. Essayez d'expliquer ce que c'est qu'un Ramdisk à un non-initié, vous allez vite comprendre. Macache bono, il revient toujours avec un autre gag, dix fois plus drôle que le mien, taratata, tu finiras par te les manger, tes crayons, Pibo!



Vous avez vu ses dessins? Vous n'imaginez pas à quel point il les chiade, parce que vous n'avez pas les originaux, mais quand je réduis un de ses dessins, j'ai le coeur qui se déchire, je pense au génocide dont nous sommes coupables, la photocopieuse Canon NP 9030 et moi, j'entends les petits cris "hiii hiii" de tous ces détails que j'écrabouille, que je compresse, que je concasse méticuleusement. Et tiens, le sourire ironique de l'ordinateur de gauche, prends-toi ça. Il est pas jouasse, le clin d'oeil rieur du bonhomme? Il se veut un 50% dans le nez? Je suis maudit, il y a des gens qui sont sur Terre pour créer, et d'autres, des cafards, ont pour mission de tout détruire, et avec le bol qui me caractérise, je fais partie de ces damnés, mais mourquoi, mourquoi?



Le plus écoeurant, c'est qu'il n'a suivi des cours de dessin que pendant deux jours, ça l'ennuyait trop. Les fées, je vous dis. Si vous avez craqué sur les dessins de Pibo, vous pouvez en trouver d'autres en feuilletant Guitar World, CD Mag (pas de chance, ils ont coulé), ou en allant au New Moon à Pigalle, à Paris.

Il a fait toutes ces illustrations (il n'appelle même pas ça des dessins, il est modeste, le bougre!) en quelques nuits, il a bossé comme un malade; on lui demandait un dessin tout simple, il nous apportait ça avec les ombres, les reflets, les trames et les machins, et nous, autour de la table lumineuse, soufflés de voir qu'il avait fait tout ça pour nous.

J'ai quand même réussi à le mettre en défaut, notre génie du crayon. Et toc. Il a confondu "astrologie" et "astronomie", comme 66.667% des français, ce qui fait que le listing "Point Astro" a une illustration qui n'a rien à voir avec le sujet, je suis bien content, bien fait bien fait!

SOMMAIRE

HOT - DOG	*STOS BASIC*	page 4
EYES	*C & 68000*	page 6
EDITEUR	*GFA BASIC*	page 8
VIRUS KICKER 2	*GFA BASIC*	page 9
INTERPOL	*GFA BASIC*	page 10
SPACE 19	*GFA BASIC*	page 11
MIDIBUF	*GFA & 68000*	page 12
BUTTON X	*68000*	page 14
POINT ASTRO	*OMIKRON*	page 15
FETCHER	*GFA BASIC*	page 16
LARGONJI	*GFA BASIC*	page 16
BOUFTOUT	*GFA BASIC*	page 21
RAMDISK	*68000*	page 24
DEAD KEYS	*68000*	page 25
KEY_DEFS	*68000*	page 26
FX80	*68000*	page 29
HARCOPY	*GFA BASIC*	page 32
GFA-PUNCHS	*GFA BASIC*	page 33

TIENS ET ÇA VOUS SAVEZ CE QUE C'EST, ÇA?

Mesdames et messieurs, veuillez faire silence et m'écouter deux minutes, venez vous asseoir autour de moi, je vais vous confier un secret, sous vos yeux, ni une ni deux, je vais casser un consensus, suivez mes mains et trouvez sous quel verre est le dé bleu, allons, qui me suit? Attention, mettez vos billets sur le bord de la table, je prends des risques énormes, vous repartirez dans deux minutes sans danger mais ma vie à moi sera changée, soyons généreux, mesdames et messieurs, j'aurai dans deux minutes la profession entière à mes baskets.

Ce que je vous ai montré, c'était un écran de surveillance d'un serveur de téléchargement, le plus ancien, le plus renommé, vous voyez de quoi je parle? Oui, voilà, celui-ci. Les animateurs constipés et pétochards peuvent voir les 120 derniers caractères saisis par chacun d'entre vous, que vous écriviez des messages en Bal, que vous discutiez en dialogue, ou que vous consultiez les petites annonces, etc. Les touches de fonctions sont indiquées, les textes, tout ce qui permet de vous suivre partout sur le service. Vous pensiez que votre correspondance ne regardait que vous, vous estimiez pouvoir vous échanger peinards des secrets et des mots doux? Vous plaisantez? Un petit intermède musical et on se retrouve.

"Fermer les volets,
Ne plus donner d'eau aux fleurs
Ne plus savoir qui tu es
Et ne plus jamais avoir peur"

Evidemment, les fournisseurs de service sont civilement responsables de tous les messages publics, cela signifie plus précisément qu'ils doivent empêcher "la provocation aux crimes et aux délits, l'offense au Président de la République (je vous jure que c'est vrai), la publication de fausses nouvelles et la diffamation". Il y a des serveurs qui font du zèle, en observant scrupuleusement chacun des connectés, basta la vie privée.

Promis, chez ST Magazine, il n'y a pas de tel contrôle, sur la vie d'ma mère, vous pouvez venir comploter et faire des misères à Tonton en paix. Par la même occasion, venez essayer notre téléchargement, c'est le plus rapide sur Minitel, il a un nom rigolo, il s'appelle Sapristi, ça ne coûte "que" 98 centimes à la minute (par opposition aux 125 centimes d'autres serveurs).

```
15 *NOUDUJ                    19:04  BAL
   95                                 19011683
... JE VOUS FELICITE POUR VOTRE TELECHA
RGEMENT.H.IL EST RAPIDE ET FIABLE C'EST
LA PREMIE.H.RE FOIS QUE J'ESSAIE > <...
----------------------------------------
16  STISTE!                   19:05  TRANSFERT
   76                                 18062583
.A.D.1.F.AUT.F.RDV.F.3.TEL.4.> <.......
.......................................
.......................................
.......................................
----------------------------------------
18 *IMPAIR                    18:55  DIA
   91                                 11018583
.IMPAIR.A.12345.A.LIR.F.DIA.H.15.SALUT!
.A.H.H.A.OUI, J'AI RECU DES NEWS TU PEU
M'APPELER AU 64 64 64 64, OK?.A.> <....
----------------------------------------
30 T3    1 T2
COMMANDE:
```

N°, pseudo, heure d'arrivée, lieu, département, sous-adresse appelante, 120 trucs.

INDEX DES ANNONCEURS

OMIKRON	page 15
MICRO-APPLICATION	en vacances
APPLICATION SYSTEMS	injoignable
HUMAN TECHNOLOGIES	trop occupé
GENERAL VIDEO	du pain sur la planche
SUD MICRO	du lait sur le feu
ULTIMA	en dérangement
ALM	sur le point de partir
AROBACE	déja parti
EXPOSE	laissez votre adresse
KOMELEC	on vous écrira
MICRO-C	écoutez je vais réfléchir
LOG ACCESS	en déplacement
MICRO-PUNCH	au lit avec 40°
UBI SOFT	mains dans le cambouis
LOGISOFT	rappelez plus tard
OCTET-PLUS	en réunion
EXTRADOS	stmagiquoi?
MICRO-VIDÉO	le bébé fait ses dents
20TH CENTURY SOFT	c'est vraiment si urgent?
COCONUT	et sinon ça va?
VORTEX	beau temps pour la pêche
SCAP	parti à l'extérieur
AMIE	en congé maternité
BASIC 1000D	croyez bien que j'aimerais
MICROTEC	c'est dur pour tout le monde
EUROMATIQUE TECH.	j'ai du bruit autour de moi
IMAGINE'S	des parasites sur la ligne
TRINOLOGY	il est en rendez-vous
POWER COMPUTING	j'aime beaucoup le n°53
EPIGRAF	je peux laisser un message
CLAVIUS	la Madame elle est pas là
MICROSPEED	ST Mag, c'est sur Mac, non?
ID	on a déjà donné
MICROSELF	et puis quoi encore?
JESSICO	nada
EUROSOFT	bernique
MUSILOG	que d'chi
VIDEOSHOP,	nib nib
ELECTRON	zarma
ESAT SOFTWARE	peau de balle
UPGRADE EDITIONS	tu l'as vu celui-la?
MICROTEC	écoutez, il n'y a personne
DAKOTA	100 balles, ça te va?

LE SERVEUR DE STMAG

C'EST 3615 STMAG

LES FICHIERS SONT DISPOS

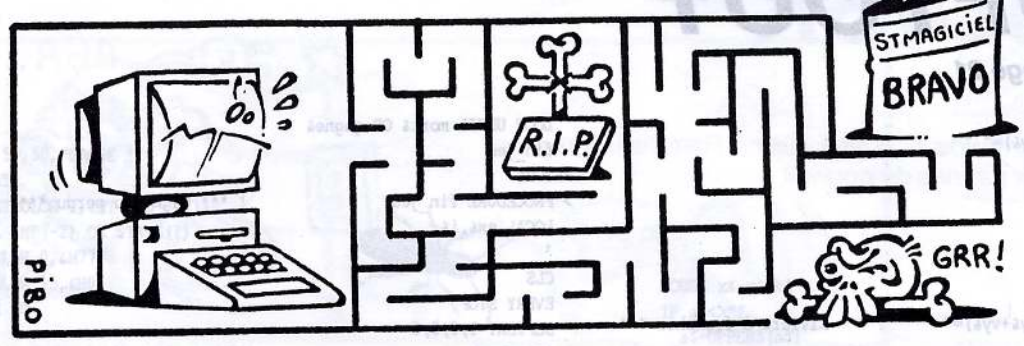
COPIEZ-LES AVEC SAPRISTI

BOUFTOUT

GFA-BASIC

Dans les jeux micro-informatiques, vous avez 90% de chances de tomber dans un labyrinthe. Ca n'a pas raté, vous voilà dans un labyrinthe, il va falloir ruser pour sortir, en prenant les clés, les bateaux et les trésors quand il faut. Vous pouvez créer vos propres tableaux.

Raphaël DELCROIX



```
ON BREAK GOSUB br
\ ***************************************************************
\ *                    BOUFTOUT                                 *
\ *              Par Raphael DELCROIX                           *
\ ***********************************************GFA_BASIC**V3.5*****
\
init_game
DO
  presentation
  RESTORE 1
2:
  cherche_niveau
  play
  IF gagne&
    GOTO 2
  ENDIF
LOOP
\ *** Jeu ***
> PROCEDURE init_game                ! *** Préparation du jeu ***
  LOCAL i&
  \
  DIM ca$(36,13)          ! * Niveau cases par cases
  DIM spr$(3)             ! * Sprites des murs
  DIM teleport&(10,2,2)   ! * Cases de téléportation(x/y départ et arrivée *10)
  DIM ca2$(13)            ! * Niveau lignes par lignes
  DIM sc$(10,2)           ! * Scores: <nom+prénom>.<score>
  HIDEM
  \ *** Dessine les murs ***
  FOR i&=3 TO 5
    DEFFILL i&
    PBOX 1,1,7,7
    GET 0,0,7,7,spr$(i&-2)
  NEXT i&
  CLS
  \ *** Lit les messages affichés en bas ***
  DIM msg$(500)
  RESTORE messages
  DO
    INC msg&
    READ msg$(msg&)
  LOOP UNTIL msg$(msg&)="*"
  CLS
  \ *** Recherche les scores ***
  nom$="A:\SCORES.BFT"
  IF EXIST(nom$)
    OPEN "i",#1,nom$
    FOR i&=1 TO 10
      INPUT #1,sc$(i&,1)
      INPUT #1,sc$(i&,2)
    NEXT i&
    CLOSE
  ENDIF
RETURN
> PROCEDURE presentation             ! *** Comme le nom l'indique... ***
  LOCAL a$,i&
  \
  tresor&=0
  tresor2&=0
  sante&=50
  temps&=0
  sc&=0
  nv&=0
  \
  WAVE 0
  SOUND 1,13,1,1
  SOUND 2,13,1,3
  SOUND 3,13,1,5
  WAVE 7
  PAUSE 50
  WAVE 0
  HIDEM                      ! *** Cache la souris ***
  OUT 4,18                   ! *** Désactive la souris ***
reprise:
  CLS
  SETCOLOR 0,&H777
  SETCOLOR 15,0
  DEFTEXT 1,1,0,26
  TEXT 75,50,"BOUF TOUT"
  DEFTEXT 1,0,0,13
  TEXT 60,80,"1. Règles"
  TEXT 60,100,"2. Jouer"
  TEXT 60,120,"3. Créer"
  TEXT 60,140,"4. Scores"
  TEXT 60,160,"5. Quitter"
  \
  \ *** Vide le buffer ***
  REPEAT
  UNTIL INKEY$=""
  \
  \ *** Attente d'une touche pressée ***
  REPEAT
    SETCOLOR 0,&H666
    SETCOLOR 0,&H555
    SETCOLOR 0,&H444
    SETCOLOR 0,&H555
    SETCOLOR 0,&H666
    SETCOLOR 0,&H777
    FOR i&=1 TO 20
      SETCOLOR 0,&H777
    NEXT i&
    a$=INKEY$
```

JEU Basse Résolution

```
UNTIL VAL(a$)<6 AND VAL(a$)>0
\
\ *** Analyse de la touche pressée ***
DEFTEXT 1,3,0,13
SELECT VAL(a$)
CASE 1
  TEXT 60,80,"1. Règles"
  PAUSE 50
  regle
  GOTO reprise
CASE 2
  TEXT 60,100,"2. Jouer"
  PAUSE 50
CASE 3
  TEXT 60,120,"3. Créer"
  PAUSE 50
  creer
  GOTO reprise
CASE 4
  TEXT 60,140,"4. Scores"
  PAUSE 50
  SETCOLOR 0,0
  aff_score2
  SETCOLOR 15,&H777
  GOTO reprise
CASE 5
  TEXT 60,160,"5. Quitter"
  PAUSE 50
  br
ENDSELECT
RETURN
> PROCEDURE cherche_niveau           ! *** Préparation du niveau ***
  LOCAL i&,n&
  \
  DIM cal$(13)
  VSETCOLOR 0,&H777
  VSETCOLOR 1,0
  VSETCOLOR 3,&H70
  VSETCOLOR 4,&H40
  VSETCOLOR 5,&H40
  CLS
  FOR i&=0 TO 7
    SETCOLOR 0,7-i&,7-i&,7-i&
    SETCOLOR 15,i&,i&,i&
  NEXT i&
  IF suite&                          ! *** Niveau à charger ***
    GOTO passe
  ELSE                               ! *** Niveau à lire en DATA ***
    READ tps$
  ENDIF
  IF tps$="*"                        ! *** Si fin des DATA ***
    CLR tps$
    IF EXIST("A:\LVL.BFT") AND NOT suite&  ! * Si niveau à charger *
      lvl_obt&=niv&                  ! *** Niveau atteint ***
      OPEN "i",#1,"LVL.BFT"
      INPUT #1,lvl_max%              ! *** Niveau maximum ***
      CLOSE
      suite&=TRUE
    ELSE                             ! *** Sinon... ***
      jeu_termine
    ENDIF
  ELSE                               ! * S'il y a encore des DATA *
    tps&=VAL(tps$)
    CLR tps$
    READ add_temps&
    FOR i&=1 TO 13
      READ cal$(i&)
    NEXT i&
  ENDIF
passe:
  IF suite& AND niv&<=lvl_max%       ! *** Chargement du niveau ***
    OPEN "i",#1,"A:\LVL_"+STR$(niv&-lvl_obt)+".BFT"
    INPUT #1,tps&
    INPUT #1,add_temps&
    FOR i&=1 TO 13
      INPUT #1,cal$(i&)
    NEXT i&
    CLOSE
  ELSE IF suite& AND niv&>lvl_max%   ! * Si dernier niveau terminé *
    jeu_termine
  ENDIF
  \
  PRINT AT(17,12);"PRET ?"
  DO
  LOOP UNTIL STRIG(1)
  CLS
  tresor&=0
  clef&=0
  bato&=0
  x&=1
  y&=7
  \
  \ *** Analyse des niveaux ***
  CLS
  ADD temps&,tps&
  DIM nb&(10)
  FOR i&=1 TO 13
    PRINT cal$(i&)
    FOR n&=1 TO 36
      ca$(n&,i&)=MID$(cal$(i&),n&,1)
      IF ca$(n&,i&)="B"
```

EMBETER STJC ? 3615 STMAG !

```
        PRINT AT(n&,i&);CHR$(241)
      ELSE IF ca$(n&,i&)="~"
        PRINT AT(n&,i&);CHR$(247)
      ELSE IF ca$(n&,i&)="*"
        PUT n&*8-8,i&*8-8,spr$(1)
      ELSE IF ca$(n&,i&)="#"
        PUT n&*8-8,i&*8-8,spr$(2)
      ELSE IF ca$(n&,i&)="/"
        PUT n&*8-8,i&*8-8,spr$(3)
        ca$(n&,i&)="*"
      ELSE IF ASC(ca$(n&,i&))<=58 AND ASC(ca$(n&,i&))>=48
        PRINT AT(n&,i&);"X"
        nn&=VAL(ca$(n&,i&))
        INC nb&(nn&)
        IF nb&(nn&)>2
          PRINT "Erreur dans le plan du niveau ";niv&+1;"."
          PRINT "(téléportage:trop)"
          PRINT "Choisissez l'option <créer>."
          ~INP(2)
          RUN
        ENDIF
        teleport&(nn&,1,nb&(nn&))=i&
        teleport&(nn&,2,nb&(nn&))=n&
      ENDIF
    NEXT n&
  NEXT i&
  FOR i&=0 TO 9
    IF nb&(i&)=1
      PRINT "Erreur dans le plan du niveau ";niv&+1;"."
      PRINT "(téléportage:pas assez)"
      PRINT "Choisissez l'option <créer>."
      ~INP(2)
      RUN
    ENDIF
  NEXT i&
  ERASE nb&(),cal$()
RETURN
> PROCEDURE play                     ! *** Jeu proprement dit ***
  LOCAL i&,var&,var1&,var2&
  LOCAL vx&,vy&,x1&,y1&,joy&
  LOCAL pos&,tr&,tax&,eau&,eaux&,eauy&
  \
  mort&=FALSE
  gagne&=FALSE
  WAVE 0
  EVERY 100 GOSUB temps
  SETMOUSE 319,199
  PRINT AT(x&,y&);CHR$(237)                  ! *** Vous... ***
  PRINT AT(1,21);msg$(INT(RND*(msg&-1))+1)   ! *** Message ***
  bloque&=0
  \
  \ *** Jeu ***
  DO
    EXIT IF ASC(INKEY$)=27                   ! *** Sortie du jeu ***
    PRINT AT(20,15);"SANTE  :";sante&;" "
    PRINT AT(20,16);"TEMPS  :";temps&;" "
    PRINT AT(20,17);"TRESOR :";tresor&;" "
    PRINT AT(20,18);"CLEFS  :";clef&;" "
    PRINT AT(20,19);"BATEAUX:";bato&;" "
    \
    \ *** Joystick géré ***
    joy&=STICK(1)
    vx&=0
    vy&=0
    SELECT joy&
    CASE 1,5,9
      vy&=-1
    CASE 2,6,10
      vy&=1
    CASE 4,5,6
      vx&=-1
    CASE 8,9,10
      vx&=1
    ENDSELECT
    IF x&+vx&<1
      vx&=0
      vy&=0
    ENDIF
    pos&=1
    IF joy&<>0                       ! *** Analyse des déplacements ***
      IF x&+vx&=37                   ! *** Mur du fond dépassé ***
        gagne&=TRUE
        GOTO pass
      ENDIF
      \
      \ *** Déplacement possible ? ***
      IF ca$(x&+vx&,y&+vy&)="*"      ! *** Mur rencontré ***
        sante&=sante&-10
        test
        FOR i&=1 TO 20
          SOUND 1,15,1,i&
          PAUSE 1
        NEXT i&
        SOUND 1,0
        pos&=0
      ELSE IF ca$(vx&+x&,vy&+y&)="-" OR ca$(x&+vx&,y&+vy&)="|"
        \                            ! *** Porte ***
        IF clef&>0
```

suite en page 22

suite de la page 21

```
      DEC clef&
      ca$(x&+vx&,y&+vy&)=" "
    ELSE
      pos&=0
      DEC sante&
      test
      PAUSE 20
    ENDIF
  ELSE IF ca$(x&+vx&,y&+vy&)="~"      ! *** Rivière à franchir ***
    IF bato&>0
      DEC bato&
      eau&=2
      eaux&=x&+vx&
      eauy&=y&+vy&
    ELSE
      pos&=0
      PAUSE 20
    ENDIF
  ENDIF
ENDIF
IF joy&<>0 AND pos&=1              ! *** Affichage du déplacement ***
  x1&=x&
  y1&=y&
  ADD x&,vx&
  ADD y&,vy&
  PRINT AT(x&,y&);CHR$(237)        ! *** Vous... ***
  IF bloque&=1                     ! *** Si téléporteur ***
    PRINT AT(x1&,y1&);"X"
  ELSE
    PRINT AT(x1&,y1&);" "
  ENDIF
  bloque&=0
  PAUSE 5
ENDIF
'
' *** Déplacement fait **
IF ca$(x&,y&)="°"                  ! *** Trésor ***
  ca$(x&,y&)=" "
  INC tresor&
  PRINT CHR$(7)
  IF tresor&>=20                   ! *** Bonus ***
    IF sante&+20>50
      tresor&=tresor&-(50-sante&)
      sante&=50
    ELSE
      ADD sante&,20
      tresor&=tresor&-20
    ENDIF
  ENDIF
ELSE IF ca$(x&,y&)="$"             ! *** Clef ***
  ca$(x&,y&)=" "
  INC clef&
  SOUND 1,15,3,3,5
  SOUND 1,0
ELSE IF ca$(x&,y&)="B"             ! *** Bateau ***
  ca$(x&,y&)=" "
  INC bato&
  SOUND 1,15,3,3,5
  SOUND 1,0
ELSE IF ca$(x&,y&)="T"             ! *** Bonus en temps ***
  ca$(x&,y&)=" "
  temps&=temps&+add_temps&
  EVERY STOP
  FOR i&=0 TO 7
    SETCOLOR 0,i&,i&,i&
    SOUND 1,15,1,i&
    PAUSE 5
  NEXT i&
  FOR i&=7 DOWNTO 0
    SETCOLOR 0,i&,i&,i&
    SOUND 1,15,1,i&
    PAUSE 5
  NEXT i&
  SOUND 1,0
  EVERY CONT
ELSE IF ASC(ca$(x&,y&))<=58 AND ASC(ca$(x&,y&))>=48 AND bloque&=0
  '                                  ! *** Téléporteur ***
  PRINT AT(x&,y&);"X"
  IF teleport&(VAL(ca$(x&,y&)),2,1)=x& AND teleport&(VAL(ca$
(x&,y&)),1,1)=y&
    var&=2
  ELSE
    var&=1
  ENDIF
  var1&=teleport&(VAL(ca$(x&,y&)),2,var&)
  var2&=teleport&(VAL(ca$(x&,y&)),1,var&)
  x&=var1&
  y&=var2&
  PRINT AT(x&,y&);CHR$(237)
  bloque&=1
ENDIF
IF eau&=2 AND joy&<>0              ! *** Affichage si rivière
***
  eau&=1
ELSE IF eau&=1 AND joy&<>0
  eau&=0
  PRINT AT(eaux&,eauy&);CHR$(247)
ENDIF
IF tresor&>70                      ! *** Amande ***
  tax&=INT(RND*20)+40
  PRINT AT(1,20);"Impots sur les grandes fortunes...-";tax&
  tr&=tresor&
  FOR tresor&=tr& DOWNTO tr&-tax&
    PRINT AT(20,17);"TRESOR :";tresor&;" "
    PRINT AT(21,21);CHR$(7)
    PAUSE 5
  NEXT tresor&
  PRINT AT(1,20);SPACE$(35)+SPACE$(LEN(STR$(tax&)))
ENDIF
pass:
LOOP UNTIL mort& OR gagne&
fin_jeu
RETURN
> PROCEDURE fin_jeu                ! *** Gagné ou perdu ***
  LOCAL ex&,i&
  '
  CLS
  EVERY STOP
  DEFTEXT 1,0,0,6
  IF gagne&                        ! *** Gagné ? ***
    INC niv&
    TEXT 10,10,"Reussite du niveau "+STR$(niv&)+":"
    DEFTEXT 1,0,0,4
    TEXT 30,30,"Sante restante..."+STR$(sante&)
    TEXT 30,50,"Temps restant...."+STR$(temps&)
    TEXT 30,70,"Tresors emportes."+STR$(tresor&)
    PAUSE 50
    ADD tresor2&,tresor&
    TEXT 35,85,"Voyons vos gains..."
    FOR i%=pts% TO pts%+(sante&+2*clef&+temps&+2*tresor&+50*niv&)
      TEXT 35,95,">"+STR$(i%)
    NEXT i%
    ADD pts%,sante&+2*clef&+temps&+2*tresor&+50*niv&
    IF temps&>0
      REPEAT
        IF sante&+1>50
          sante&=50
        ELSE
          ADD sante&,1
        ENDIF
        TEXT 30,30,"Sante restante..."+STR$(sante&)
        DEC temps&
        TEXT 30,50,"Temps restant...."+STR$(temps&)+" "
        PRINT AT(1,1);CHR$(7)
        PAUSE 5
      UNTIL temps&=0 OR sante&=50
    ENDIF
    REPEAT
    UNTIL INKEY$<>"" OR STRIG(1)
  ELSE                             ! *** Perdu ? ***
    TEXT 10,10,"Perdu..."
    TEXT 30,30,m$
    m$=""
    REPEAT
    UNTIL INKEY$<>"" OR STRIG(1)
    '
    ' *** Scores ***
    IF pts%>0
      look_score
    ENDIF
    aff_score2
    CLS
    FOR i&=7 DOWNTO 0
      SETCOLOR 0,7-i&,7-i&,7-i&
      SETCOLOR 15,i&,i&,i&
    NEXT i&
  ENDIF
RETURN
> PROCEDURE temps                  ! *** Routine de minutage ***
  DEC temps&
  IF temps&<21
    SOUND 1,15,1,3,2
    SOUND 1,0
  ENDIF
  IF temps&<0
    temps&=0
    mort&=TRUE
    m$="Le temps est VOTRE MAITRE..."
    SOUND 1,15,4,1,20
    SOUND 1,0
  ENDIF
RETURN
> PROCEDURE test                   ! *** Test de santé ***
  IF sante&<0
    m$="Ce n'etait pas la forme..."
    sante&=0
    mort&=TRUE
  ENDIF
RETURN
'
> PROCEDURE jeu_termine
  LOCAL rep&
  '
  gagne
  SHOWM
  CLS
  PRINT "BONUS: ";+tresor2&*2;" POINTS !!!"
  ADD pts%,tresor2&*2
  PRINT "POINTS OBTENUS: ";pts%
  DO
  LOOP UNTIL INKEY$<>"" OR STRIG(1)
  look_score
  aff_score2
  WAVE 0
  OUT 4,8
  SHOWM
  ALERT 0,"Choisissez:",1,"ENCORE|STOP|CREE",rep&
  HIDEM
  OUT 4,18
  HIDEM
  IF rep&=1
    RUN
  ELSE IF rep&=2
    br
  ELSE
    creer
    RUN
  ENDIF
RETURN
> PROCEDURE look_score             ! *** Scores battus ? ***
  LOCAL ex&,i&,nb&,nb2&
  '
  FOR nb&=1 TO 10
    IF pts%>=VAL(sc$(nb&,2))
      ex&=TRUE
      FOR i&=9 DOWNTO nb&
        sc$(i&+1,1)=sc$(i&,1)
        sc$(i&+1,2)=sc$(i&,2)
      NEXT i&
      sc$(nb&,2)=STR$(pts%)
      aff_score1
      OPEN "o",#1,nom$
      FOR nb&=1 TO 10
        PRINT #1,sc$(nb&,1)
        PRINT #1,sc$(nb&,2)
      NEXT nb&
      CLOSE
    ENDIF
  NEXT nb&
RETURN
> PROCEDURE aff_score1             ! *** Demande d'entrée du nom ***
  LOCAL t$,i&,j&
  '
  CLS
  VSETCOLOR 0,0
  VSETCOLOR 1,&H777
  FOR i&=2 TO 7
    VSETCOLOR i&,7,(i&-1),(i&-1)
  NEXT i&
  FOR i&=8 TO 11
    VSETCOLOR i&,7,(6-(i&-7)),(6-(i&-7))
  NEXT i&
  VSETCOLOR 12,0
  VSETCOLOR 13,112
  '
  DEFTEXT 1,1,,20
  TEXT 15,20,"YOU'RE ONE OF THE BEST"
  c&=1
  nb2&=0
  DEFTEXT ,,,6
  FOR i&=70 TO 160 STEP 10
    INC c&
    INC nb2&
    DEFTEXT c&
    TEXT 20,i&,-50,nb2&
    IF nb2&=nb&                    ! *** Nom+prénom à marquer ***
      j&=i&
    ELSE
      IF sc$(nb2&,2)<>""
        TEXT 50,i&,-200,sc$(nb2&,1)
        TEXT 270,i&,-30," "+sc$(nb2&,2)
      ELSE
        TEXT 50,i&,"************************"
        TEXT 270,i&,-30," X"
      ENDIF
    ENDIF
  NEXT i&
  TEXT 270,j&,-30," "+sc$(nb&,2)
  DO
    a$=INKEY$
    IF LEN(t$)=25
      a$=CHR$(13)
    ENDIF
    IF a$<>CHR$(13) AND a$<>""
      IF ASC(a$)=8 AND LEN(t$)>0   ! *** Touche <Backspace> ***
        a$=""
        t$=LEFT$(t$,LEN(t$)-1)
      ELSE
        t$=t$+UPPER$(a$)
      ENDIF
      TEXT 50,j&,t$+SPACE$(25-LEN(t$))
    ENDIF
  LOOP UNTIL a$=CHR$(13) AND t$<>""
  sc$(nb&,1)=t$
  '
RETURN
> PROCEDURE aff_score2             ! *** Affiche les meilleurs scores ***
  LOCAL i&,c&,nb&,a&,pal$
  '
  CLS
  VSETCOLOR 1,&H777
  FOR i&=2 TO 7
    VSETCOLOR i&,7,(i&-1),(i&-1)
  NEXT i&
  FOR i&=8 TO 11
    VSETCOLOR i&,7,(6-(i&-7)),(6-(i&-7))
  NEXT i&
  VSETCOLOR 12,0
  VSETCOLOR 13,112
  '
  DEFTEXT 1,1,,20
  TEXT 50,20,"THE BEST PLAYERS"
  c&=1
  nb&=0
  DEFTEXT ,,,6
  FOR i&=70 TO 160 STEP 10
    INC c&
    INC nb&
    DEFTEXT c&
    TEXT 20,i&,-50,nb&
```

EMBETER STJC ?
3615 STMAG !

suite en page 23

Page manquante

RAMDISK

68000 & GFA

Stockez (provisoirement) vos données sur ce Ramdisk. Il résiste au Reset, mais pas aux pannes de courant...

Eric DA CUNHA

```
*************************************
* Ramdisk en boot resistant au reset*
*    COPYRIGHT DA CUNHA ERIC 1990   *
*************************************
;  Utiliser le programme joint en Gfa pour
;  l'installer sur un disk

; Les valeurs par défaut correspondent a un 2.5 M
; supporte les capacités suivantes:
;  +128 K pour 520  ST?
;  +256 K pour 1040 ST?
;  +448 K pour 2.5 M ou 2M ou 4M

unit      equ        'C'-65  ;unités 'C' à 'H'

          cmp.b      #$2a,$fffffc02.w
;shift gauche pour ne pas installer ramdisk
          bne.s      ok
          rts
ok        move.l     $42e.w,d0
          LEA        start(PC),A1
          cmp.l      #$200001,d0
;$180001 pour 2M ;$380001 pour 4M
          bhi.s      cop
          moveq      #$c,d1
          lea        mes2(pc),a0
          cmp.l      #$080000,D0
          BHI.s      dif
L520      moveq      #6,d1
          lea        mes3(pc),a0
dif       MOVE.W     (A0)+,c0+2-start(A1)
          move.b     d1,c1+3-start(a1)
          move.b     d1,c2+3-start(a1)
          move.b     d1,C3+3-start(a1)
          move.b     d1,hdv_bpb+3-start(a1)
          move.l     (a0)+,C4-start(a1)
          move.w     (a0),C5-start(a1)
cop       pea        mes(pc)
          move.w     #9,-(a7)
          trap       #1
          addq.l     #6,a7
          LEA        $2c0.w,A0
          MOVEQ      #(end-start)/4,D1
rc        MOVE.L     (A1)+,(A0)+
          DBF        D1,rc
C0        MOVE.W     #$1C0,D7
c2        lea        $210000,a6
;$190000 pour 2M ;$390000 pour 4M
C1        CMP.L      #$210000,$42E.W
;idem
          BEQ.S      set
          MOVEQ      #0,D1
          LEA        (A6),A0
          MOVE.W     #$8FF,D0
pliz      MOVE.L     D1,(A0)+
          DBF        D0,pliz
          MOVE.L     A6,$42E.W
          move.l     4.w,a0
          jmp        (a0)
set       MOVE.L     $472.w,$2c8+2.W
          MOVE.L     #$2c0,$472.w
          MOVE.L     $476.W,$2de+2.W
          MOVE.L     #$2d6,$476.W
          MOVE.L     $47E.W,$366+2.W
          MOVE.L     #$35e,$47E.W
          LEA        $C(A6),A0
          LEA        BINF(pc),A1
          MOVE.W     D7,d6
          LSL.W      #1,D7
          ADD.W      #$12,D7
          MOVE.B     D7,$13-$C(A1)
          LSR.W      #8,D7
          MOVE.B     D7,$14-$C(A1)
          MOVEQ      #15+14,D0
rc2       MOVE.B     (A1)+,(A0)+
          DBF        D0,rc2
          move.w     d6,(a0)+
          Bset       #UNIT,$4C5.W
          RTS
mes       dc.b       "Edy's "
C4        dc.b       "448K Ramdisk->High=$"
C5        dc.b       "210000!",0
;$190000 pour 2M ;$390000 pour 4M
          even
mes2      DC.W       256
          dc.b       "256K","0C"
mes3      DC.W       128
          dc.b       "128K","06"
BINF      dc.b       2,2,1,0,2,$70,0,2,0,0,5,0,9,0,1,0
          DC.B       2,0,0,2,4,0,0,7,0,5,0,6,0,$12
start     CMP.W      #unit,4(A7)
          BEQ.S      hdv_bpb
          jmp        $11111111
hdv_bpb   MOVE.L     #$21001C,D0
;$19001c pour 2M ;$39001c pour 4M
          RTS
hdv_rw    CMP.W      #unit,$E(A7)
          BEQ.S      wrea
          jmp        $22222222
wrea      MOVE.W     $C(A7),D0
          EXT.L      D0
          moveq      #9,d1
          lsl.l      d1,d0
          MOVEA.L    6(A7),A0
          MOVE.W     $A(A7),D1
          SUBQ.W     #1,D1
C3        LEA        $210000,A1
;$190000 pour 2M ;$390000 pour 4M
          ADD.L      D0,A1
          MOVE.W     4(A7),D0
          BTST       #0,D0
          BEQ.S      read
          EXG        A1,A0
read      MOVE.W     A0,D0
          BTST       #0,D0
          BNE.S      imp
          MOVE.W     A1,D0
          BTST       #0,D0
          BNE.S      imp
bloc0     MOVEQ      #$A,D0
bloc      MOVEM.L    (A1)+,D2-d7/A2-a6
          MOVEM.L    D2-d7/A2-a6,(A0)
          LEA        $2C(A0),A0
          DBF        D0,bloc
          MOVEM.L    (A1)+,D2-d7/A2
          MOVEM.L    D2-d7/A2,(A0)
          LEA        $1C(A0),A0
          DBF        D1,bloc0
          bra.s      imp2
imp       MOVEQ      #63,D0
oct       rept       8
          MOVE.B     (A1)+,(A0)+
          endr
          DBF        D0,oct
          DBF        D1,imp
imp2      MOVEQ      #0,D0
          RTS
hdv_mediach:
          CMP.W      #unit,4(A7)
          BEQ.S      imp2
          jmp        $33333333
end
```

```
' Programme GfA:
' Installe l'exécutable du listing ramdisk en boot
' Copyright 1990  Da Cunha Eric pour ST MAG
a$=SPACE$(512)
~XBIOS(8,L:V:a$,L:-1,W:0,W:1,W:0,W:0,W:1)
' lit le bootsecteur courant
ch=0
FOR i=0 TO 511 STEP 2
  ch=ch+DPEEK(V:a$+i)
  ch=ch MOD &H10000
NEXT i
' calcule son checksum
IF ch=&H1234
  PRINT "Boot exécutable"
  INPUT "Poursuivre ",r$
  IF UPPER$(r$)<>"O"
    END
  ENDIF
ENDIF
b$=SPACE$(512)
BLOAD "a:boot.dat",V:b$
BMOVE V:b$+28,V:a$+30,508-28
DPOKE V:a$,&H601C
LPOKE V:a$+2,CVI("Edy!")
ch=0
FOR i=0 TO 509 STEP 2
  ch=ch+DPEEK(V:a$+i)
  ch=ch MOD &H10000
NEXT i
DPOKE V:a$+510,(&H1234-ch) MOD &H10000
' prépare le nouveau boot exécutable
ch=0
FOR i=0 TO 511 STEP 2
  ch=ch+DPEEK(V:a$+i)
  ch=ch MOD &H10000
NEXT i
~XBIOS(9,L:V:a$,L:-1,W:0,W:1,W:0,W:0,W:1)
' ecrit le résultat
' fini .......
```

UTILITAIRE
Toutes Résolutions

BOUFTOUT

suite de la page 23

```
DATA Et puis zut !
DATA J'suis à cours d'idée !
DATA Bon c'est pas tout mais j'vous quitte.
DATA *
' *** Niveaux ***
1:
DATA 120,30
DATA ************************************
DATA * §*   *°*§*B*  °*°*T*   *°*       *
DATA * ** *   * *   * * * * * * * ***** *
DATA *    * *   * *** * * * * *§*     * *
DATA ****** ***** *   * * * *   * ***~*°*
DATA * *°*  *§ °° * *** *_*_***** *   ***
DATA   * * *** ** * * | | | ***** * *   |
DATA * * * *§* ** * * * * *   °   *°*****
DATA *   *     ** * * * * ***********   *
DATA *** * *** ** * * * *     °       * *
DATA *   * *   **_* * * *************** *
DATA *§*   *°*°**T*   *       °         *
DATA ************************************
' 2
DATA 50,100
DATA ************************************
DATA *        **   ********           **
DATA *~********§** * **°°°°** ******** **
DATA *   *********** ***** **B*§§§~   **
DATA ***            **    ** ***********
DATA *** ******* *****BB** ** *****  ||||
DATA   * ***   * ********* **~*****_****
DATA *    ** *§* §      |           ****
DATA * ** **_*§*_********** *************
DATA * *****-*§*°°°°°**B*** *°°§§§|T    *
DATA *       ***°*°*°°*°*** *********** *
DATA ***°***   |°*°**°°°***        |§BB§B*
DATA ************************************
' 3
DATA 50,20
DATA ************************************
DATA *  #°°°°°°°°°°°*°°°°°°°°°°°°°°°°°*
DATA * * **********°***°*************°*
DATA * * *°°°°°°°°°°*°*°*°°°°°°°°°°°°*°*
DATA * * *°**********°#°*************°*°*
DATA * * *°°°°°°°°°°°°*°/   B§    /°*°/
DATA   * ****************#**********#*#*#*
DATA * * *            *   *********T|   *
DATA * * * ********** #T*********** *****
DATA * * *          * * #     ~         #
DATA * * ********** *#* *************** /
DATA *   #        /T*        ~          *
DATA ************************************
' 4
DATA 20,50
DATA ************************************
DATA *°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°#
DATA *°**********************************
DATA *°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°*
DATA *********************************** *
DATA *°°°°°*°°°°*°°°°°*  °             *
DATA  °°°°°°°°°°°°°°°***#***************
DATA */////#////#/////***T|        |   /
DATA *    * #/     /B# *#***************
DATA *°§* * ***/*****#* *      ~       *
DATA **** * * *#*§*B*T/ *************** /
DATA *°°°°°°°°°°°°°°°*         ~       *
DATA ************************************
' 5
DATA 50,10
DATA ************************************
DATA *              *#§*       *       *
DATA * *#*/*/*#*/*/*#/* ****** * ****** *
DATA * *°*°*°*°*°*°*##*          * ****** *
DATA * *§*§#§#§#§**#* ****** * *       *
DATA * *************#*          * * ******
DATA   ~          *  *#|#******#~T* ||||||
DATA * ***_***_***  *#*          * * ******
DATA * *####*####** ##* ****** * *      *
DATA * *§*§#*#BB#*  *#*       * ****** *
DATA * ***********  *#* ****** * ****** *
DATA *              *B*        *      *
DATA ************************************
' 6
DATA 50,10
DATA ************************************
DATA *3|  | *       ///####             *
DATA ****_*   *********#**°°*********** *
DATA *1** * *_*°*B*°*6*#**# *2          *
DATA *§3*4|5*    °°°°°/#//5/************
DATA ******************#*****************
DATA                 B                 4|
DATA ******************~*****************
DATA *BB             # *6    ##* ~# *2*
DATA *******************        /1* ** * *
DATA *§/#                *#********* §* * *
DATA *###***************            °/   *
DATA ************************************
' 7
DATA 50,10
DATA ************************************
DATA *°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°1*
DATA *////////////////~////////////////*
DATA *                                **
DATA *§***************/*****************
DATA **** 1*           * # *          **
DATA  0*0 2*           * 3 *       *****
DATA **** 3*           * # *       |B5|~
DATA *4**B**           **#**       *****
DATA *~****************####**************4|
DATA *                 5****/*         §**
DATA *§                 ~               2*
DATA ************************************
' 8
DATA 50,200
DATA ************************************
DATA *1*// ##/*B*§B    |   *7*°°°°°°°*6*
DATA * *1# /°2* ***********_* *       * *
DATA * ********* *              * *°°°°°°°* *
DATA *0          *2*°*   *°*3*4*        *5*
DATA *****~************~**************~******
DATA                 B                    |
DATA *****~************~********************
DATA *B*§*  *   *5 *     *6*§######4///*
DATA * *0~     * *  *°°°°°°* *//~///T////*
DATA * ******** *  *      * *//##/3/////*
DATA *             *B§*°°°°°°*7*///###/////*
DATA ************************************
' 9
DATA 50,10
DATA ************************************
DATA *  ~     |                    |    ~  *
DATA *  **    *********************    **_*
DATA *  ****   *****************    ****  *
DATA *  */// **   ############    **###*  *
DATA *  *//##***   **B**B**    ***##*#*  *
DATA    ###/######   #//#   ######//##  |
DATA *  */###***   **B**B**    ***##*/*  *
DATA *  *///**   ////////////    **##§*  *
DATA *  ****   *****************    ****  *
DATA *  **    *********************    **_*
DATA *  ~     |                    |    ~ *
DATA ************************************
' 10
DATA 50,10
DATA ************************************
DATA *B/       ~         45~            | *
DATA *#/ *****************************§* *
DATA *         ~      §  03~            | *
DATA *** ******************************** *
DATA * |       ~      °  13~            | *
DATA  §********************************* |
DATA * |       ~      °  02~            | *
DATA *** ******************************** *
DATA *         ~      B  21~            | *
DATA *#/ ********************************* *
DATA *B/       ~         54~            | *
DATA ************************************
'
DATA *
```

CA SE CROIT MALIN !

L'émulateur TT sur MAC, c'était un joke, voyons! Vous avez été nombreux à nous demander des renseignements, arrêtons les frais: le soft Mac Step n'existe pas, nous avons juste transféré des écrans TT sur Mac, c'est tout, si on peut même plus rigoler...

CRIS DU COEUR

J'ai placé un magnéto dans les locaux de ST Mag, voici quelques cris du coeur de la Rédaction:

"Purée, merde, il a encore bombé ce con!"

"Il fait chaud, j'ai chaud, j'ai vraiment super chaud, c'est la chaleur de la reumo."

"Quel est le con qui a inventé le ST et ses foutus 8 caractères pour le nom de mon fichier?"

"J'ai l'impression qu'on se fout de ma gueule."

"Purée, merde, ce con, purée, je sauve en LIB et il bombe!"

"J'ai chaud."

"Il va clamser, contaminé, il va clamser, contaminé."

"Jamais tu les fais droits, tes gabarits, Mireille?"

"On est en retard. Je le crois pas, on est en retard."

JUST CALL ME STUMPF

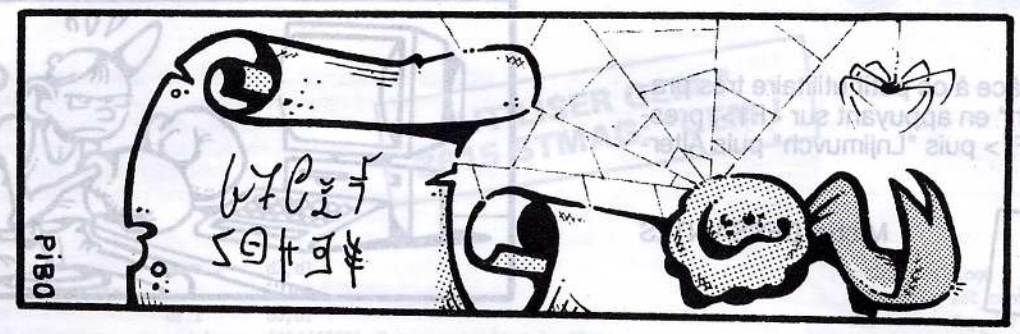
Quelque part dans le canard (phrase révélatrice d'un magazine mal foutu, réalisé par une équipe dépourvue du moindre petit chouia d'organisation), on vous a dit que Stumpf, PDG d'Atari Allemagne, était devenu Führer von Atari Welt par la volonté du Saint Esprit et aussi un peu par celle de Tonton Tramiel. On a oublié de vous dire que par conséquent il devenait aussi PDG d'Atari France, secondé par Ali Hammaoui, Directeur Général quand même.

DEAD KEYS

68000

Gräcê à cë pêtït ùtilîtàîrê bien pratique, vous aurez enfin accès aux caractères les plus bizarroïdes de votre clavier.

Michel DESSAINTES



UTILITAIRE
Toutes Résolutions

PARLER DE C ?
3615 STMAG: *CCC.

```
        plen    63
        llen    132
        ttl     GESTION DES TOUCHES MORTES
        subttl
        listchar 15
        format  0+,1+,2+
        opt     A+,M+,O+,P-,S-,T+,W+

;-----------------------------------------------------------
;
;       DEADKEYS.L Gestion des touches mortes.
;
; Auteur   : Michel DESSAINTES
;
;-----------------------------------------------------------
;       Macros DOS
;-----------------------------------------------------------

bconin:  macro   device
         move    \1,-(sp)
         move    #2,-(sp)
         trap    #13
         addq    #4,sp
         endm

getshift: macro  modus
         move    \1,-(sp)
         move    #$0b,-(sp)
         trap    #13
         addq    #4,sp
         endm

keepproc: macro  error_nr,length
         move    \1,-(sp)
         move.l  \2,-(sp)
         move    #$31,-(sp)
         trap    #1
         endm

printline: macro ^string
         pea     \1
         move    #9,-(sp)
         trap    #1
         addq.l  #6,sp
         endm

setexec: macro   ^new_address,vector_nr
         pea     \1
         move    \2,-(sp)
         move    #5,-(sp)
         trap    #13
         addq.l  #8,sp
         endm

;-----------------------------------------------------------
start_prg:
;-----------------------------------------------------------
         bra     install         ; installe le programme
                                 ; comme résident
new_trap:
         btst    #5,(a7)         ; mode système ?
         bne.s   sys_sp          ; oui
         move.l  usp,a0          ; non, prendre le user
                                 ; stack pointer
         bra.s   check_bconin
sys_sp:
         lea     6(a7),a0        ; mode système, prendre le
                                 ; system stack pointer
check_bconin:
         cmp.l   #$20002,(a0) ; dev=CON:, fonction=Bconin
         beq.s   new_bconin      ; fonction Bconin BIOS

         cmp.l   #$10002,(a0) ; dev=CON:, fonction=Bconstat
         bne     old_trap        ; ni Bconin, ni Bconstat

new_bconstat:                            ; nouveau BCONSTAT
         tst.l   pending_keys ; touches en attente à retourner?
         beq     old_trap        ; non, donc ancien Bconstat

         moveq   #-1,d0          ; affirme qu'il y a un
                                 ; caractère disponible
         bra     end_prog_rte ; retourne "pas de touche
                                 ; disponible"
new_bconin:                      ; nouveau BCONIN
         move.l  old_trap+2,a0   ; sauve nouveau vecteur de
trap
         setexec (a0),#45        ; restaure ancien vecteur
         move.l  d0,current_trap ; garde le vecteur
                                 ; trap 13 courant

         tst.l   pending_keys    ; des touches en attente ?
         beq.s   no_pending_key  ; non
return_pending_key:
         move.l  pending_keys,d0 ; oui, on retourne la 1ère
move.l   pending_keys+4,pending_keys   ; on décale la suivant
                                 ; s'il y en a une
         clr.l   pending_keys+4  ; marque qu'on l'a
                                 ; retournée
         bra.s   end_prog        ; et on la retourne

no_pending_key:                  ; aucune touche en attente
         bconin  #2              ; on lit une touche
         bsr.s   is_compose      ; caractère de composition ?
         beq.s   compose         ; oui
         bsr.s   is_auto_compose ; non; caractère de
                                 ; composition automatique ?
         beq.s   auto_compose ; oui
         bra.s   end_prog        ; non, on retourne la touche
                                 ; telle quelle
compose:                         ; la touche de composition a été entrée
         bconin  #2              ; on prend le 1er caractère
auto_compose:                    ; on arrive ici en cas de
                                 ; composition automatique
         bsr.s   check_key_1     ; test si le caractère peut
                                 ; commencer une composition
         bne.s   return_pending_key ; non, on le retourne
         bconin  #2              ; oui, alors on lit
                                 ; le 2nd caractère
         bsr     check_key_2     ; les 2 caractères forment-ils
                                 ; ils une séquence de composition?
         bne.s   return_pending_key ; non, alors, on
                                 ; retourne les 2 caractères

         clr.l   pending_keys    ; oui, on nettoie la
                                 ; touche en attente
         clr.l   pending_keys+4  ; et le dernier
                                 ; caractère entré
         move.b  3(a0),d0        ; et on retourne le
                                 ; caractère composé
end_prog:
         move.l  d0,-(a7)        ; sauve le registre D0
         move.l  current_trap,a0
         setexec (a0),#45        ; restaure le vecteur courant
         move.l  (a7)+,d0        ; restaure le registre D0
end_prog_rte:
         rte                     ; retourne du TRAP 13
;-----------------------------------------------------------
; teste si le caractère est celui de composition
is_compose:
;-----------------------------------------------------------
         cmp.b   compose_char,d0 ; c'est ce caractère ?
         bne.s   ic_rts ; non, on retourne le flag Z à zéro

         move.l  d0,-(a7)        ; oui
         getshift #-1            ; lit l'état des touches
                                 ; shift/alt/control
         move.b  d0,d1           ; sauve l'état dans D1
         move.l  (a7)+,d0        ; restaure D0
         and.b   #15,d1          ; masque les bits
                                 ; non souhaités
         cmp.b   compose_shift,d1 ; teste si l'état
                                 ; est correct
ic_rts:                          ; retourne l'état des
                                 ; touches shift/alt/control
         rts

;-----------------------------------------------------------
; teste si c'est un caractère de composition automatique
is_auto_compose:
;-----------------------------------------------------------
         lea     auto_compose_tab,a0 ; teste tous les
                                 ; caractères de la table
nxt_auto:
         tst.b   (a0)            ; fin de la table ?
         beq.s   iac_rts_n       ; oui, on retourne "pas un
                                 ; caractère automatique"
         move.b  (a0)+,d1        ; non, on prend le caractère
                                 ; dans la table
         cmp.b   d0,d1           ; le même que celui saisi ?
         bne.s   nxt_auto        ; non, teste le suivant
iac_rts_y:                       ; oui, retourne le flag Z à 1
         rts
iac_rts_n:
         tst.b   d0              ; positionne le flag Z à "pas
                                 ; un caractère automatique"
         rts

;-----------------------------------------------------------
; teste si le 1er caractère commence un séquence de composition
check_key_1:
;-----------------------------------------------------------
         move.l  d0,pending_keys ; sauve le caractère
                                 ; si ce n'est pas le cas
         move.b  d0,previous_key ; sauve le caractère
                                 ; pour "check_key_2"
         lea     conversion_table-4,a0 ; juste avant la
                                 ; table de conversion
         move    #(ct_end-conversion_table)/4-1,d1 ; sa
                                 ; longueur
nxt_ck1a:
         addq    #4,a0           ; saute à la séquence suivante
         cmp.b   (a0),d0         ; idem au caractère entré ?
         beq.s   nxt_ck1b        ; oui, on l'a trouvé
         cmp.b   1(a0),d0        ; teste le 2d caractère
         dbeq    d1,nxt_ck1a     ; différent, teste l'entrée

nxt_ck1b:
         rts                     ; retourne le flag Z (=1 si le caractère
                                 ;                    est dans la table)
;-----------------------------------------------------------
; teste si les 2 caractères sont une séquence composée
check_key_2:
;-----------------------------------------------------------
         move.l  d0,pending_keys+4 ; sauve le caractère si
                                 ; pas une séquence
         move.b  previous_key,d2 ; on prend le caractère
                                 ; précédent
         cmp.b   d0,d2           ; met le plus petit
                                 ; dans les bits 15-8
         bls.s   nxt_ck2a
         exg     d0,d2
nxt_ck2a:                        ; d2 = bits 15-8, d0 = bits 7-0
         lsl.w   #8,d2           ; le place dans l'octet
                                 ; fort du mot faible
         move.b  d0,d2           ; place le dernier
                                 ; caractère en bas

         lea     conversion_table-4,a0 ; juste avant
                                 ; la table de conversion
         move    #(ct_end-conversion_table)/4-1,d1 ; sa
                                 ; longueur
nxt_ck2b:
         addq    #4,a0           ; saute à l'entrée suivante
         cmp.w   (a0),d2         ; teste si les 2 caractères
                                 ; forment un séquence composée
         dbeq    d1,nxt_ck2b     ; si non, on teste l'entrée
                                 ; suivante
         rts                     ; retourne le flag Z (=1 si la séquence
                                 ;                    est dans la table)
;-----------------------------------------------------------
;       data
;-----------------------------------------------------------
; définition du caractère de composition
compose_shift:   dc.b    8       ; défaut = Alternate
compose_char:    dc.b    $1b     ; défaut = ESCape

pending_keys:    dc.l    0,0     ; prochaine touches à
                                 ; retourner (0 si aucune)
previous_key:    dc.b    0       ; 1er caractère de la
                                 ; séquence
auto_compose_tab: dc.b           "^¨",0,0,0,0,0  ; maximum 6
                                 ; (+ 0 à la fin)

;-----------------------------------------------------------
conversion_table:   ; table de conversion
;-----------------------------------------------------------
; entrée de table: 1er caractère, 2d caractère, 0, code d'équivalence
;        (le 1er caractère DOIT être < 2d)
; examples :
;        pour obtenir le signe "cent" ($9B),
;        entrer le caractère de composition (Alt+ESC),
;                puis "c",
;                puis "/" ou "|";
;        pour obtenir la fraction "un demi" ($AB),
;        entrer le caractère de composition (Alt+ESC),
;                puis "1",
;                puis "2";
;        pour obtenir le signe "a umlaut"  ($84),
;        ne pas entrer le caractère de composition (Alt+ESC),
;                entrer ":" ou "¨",
;                puis "a".
         even
         dc.b    "^¨",0,"^"      ; accent circonflexe
                 ; (quand on ne veut pas l'automatisme)
         dc.b    ",C",0,$80      ; C cédille
         dc.b    "u¨",0,$81,'¨u',0,$81 ; u trèma
         dc.b    "`e",0,$82      ; e accent aigu
         dc.b    "^a",0,$83      ; a accent circomflèxe
         dc.b    "a¨",0,$84,'¨a',0,$84 ; a tréma
         dc.b    "`a",0,$85      ; a accent grave
         dc.b    "a°",0,$86,'ao',0,$86 ; a ring
         dc.b    ",c",0,$87      ; c cédille
         dc.b    "^e",0,$88      ; e accent circonflexe
         dc.b    "e¨",0,$89,'¨e',0,$89 ; e trèma
         dc.b    "`e",0,$8A      ; e accent grave
         dc.b    "i¨",0,$8B,'¨i',0,$8B ; i trèma
         dc.b    "^i",0,$8C      ; i circomflèxe
         dc.b    "`i",0,$8D      ; i accent grave
         dc.b    "A¨",0,$8E,'¨A',0,$8E ; A trèma
         dc.b    "A°",0,$8F,'Ao',0,$8F ; A ring

         dc.b    "`E",0,$90      ; E accent aigu
         dc.b    "ae",0,$91      ; ligature a e
         dc.b    "AE",0,$92      ; ligature A E
         dc.b    "^o",0,$93      ; o accent circonflexe
         dc.b    "o¨",0,$94,'¨o',0,$94 ; o trèma
         dc.b    "`o",0,$95      ; o accent grave
         dc.b    "^u",0,$96      ; u accent circonflexe
         dc.b    "`u",0,$97      ; u accent grave
         dc.b    "y¨",0,$98,'¨y',0,$98 ; y trèma
         dc.b    "O¨",0,$99,'¨O',0,$99 ; O trèma
         dc.b    "U¨",0,$9A,'¨U',0,$9A ; U trèma
         dc.b    "/c",0,$9B,'c|',0,$9B ; cent
         dc.b    "-L",0,$9C      ; livre (pound)
         dc.b    "-Y",0,$9D      ; yen
         dc.b    "ss",0,$9E      ; béta
         dc.b    "-f",0,$9F      ; Florin
```

NDC: Bonjour les TABs! Je meurs!

suite en page 28

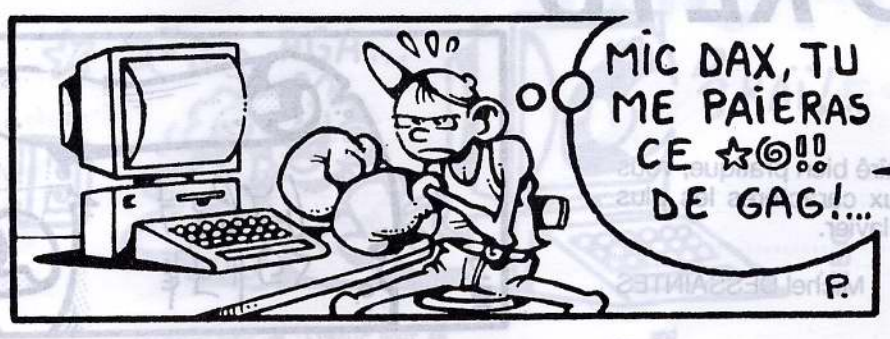
KEY_DEFS

68000

Redéfinissez vos touches grâce à ce petit utilitaire très pratique. Pour écrire "Lnjimuvch" en appuyant sur <F1>, pressez Alternate + Enter puis <F1> puis "Lnjimuvch" puis Alternate + Enter.

Michel DESSAINTES

UTILITAIRE
Toutes Résolutions



```
        plen    63
        llen    132
        ttl     GESTION DE DÉFINITIONS DE TOUCHES
        subttl
        listchar 15
        format  0+,1+,2+
        opt     A+,M+,O+,P-,S-,T+,W+

; ---------------------------------------------------------------
;
;       KEY_DEFS.S Gestion de définition des touches.
;
; Auteur : Michel DESSAINTES
;
; ---------------------------------------------------------------

conterm   = $484      ; variable BIOS
MAX_DEFS  = $400      ; taille de la zone de définition
          ;                   (en nombre de touches)
MAX_KEYS  = $40       ; taille de la zone des touches
          ;                   (en nombre de touches)
DEF_KEY   = $0872000d ; touche qui démarre et finit une def.

        subttl  MACROS
; ---------------------------------------------------------------
;       Macros DOS
; ---------------------------------------------------------------

bconin:   macro   device
          move    \1,-(sp)
          move    #2,-(sp)
          trap    #13
          addq    #4,sp
          endm

bconout:  macro   device,char
          move    \2,-(sp)
          move    \1,-(sp)
          move    #3,-(sp)
          trap    #13
          addq    #6,sp
          endm

bconstat: macro   device
          move    \1,-(sp)
          move    #1,-(sp)
          trap    #13
          addq    #4,sp
          endm

dosound:  macro   ^sound
          pea     \1
          move    #$20,-(sp)
          trap    #14
          addq.l  #6,sp
          endm

keepproc: macro   error_nr,length
          move    \1,-(sp)
          move.l  \2,-(sp)
          move    #$31,-(sp)
          trap    #1
          endm

keytbl:   macro   \unshift,\shift,\capslock
          move.l  \3,-(sp)
          move.l  \2,-(sp)
          move.l  \1,-(sp)
          move    #16,-(sp)
          trap    #14
          add     #14,sp
          endm

physbase: macro
          move    #2,-(sp)
          trap    #14
          addq    #2,sp
          endm

printline: macro  ^string
          pea     \1
          move    #9,-(sp)
          trap    #1
          addq.l  #6,sp
          addq.l  #6,sp
          endm

setexec:  macro   ^new_address,vector_nr
          pea     \1
          move    \2,-(sp)
          move    #5,-(sp)
          trap    #13
          addq.l  #8,sp
          endm

supexec:  macro   ^routine
          pea     \1
          move    #38,-(sp)
          trap    #14
          addq    #6,sp
          endm

term_2:   macro   error_nr
          move    \1,-(sp)
          move    #$4c,-(sp)
          trap    #1
          endm
```

TELECHARGEMENT 3615 STMAG !

```
; ---------------------------------------------------------------
;       Macros Internes
; ---------------------------------------------------------------

signal_start: macro
          dosound   snd_start
          endm

signal_end: macro
          dosound   snd_end
          endm

signal_fault: macro
          dosound   snd_fault
          endm
          subttl    TEST APPEL AU BIOS
          page
; ---------------------------------------------------------------
start_prg:
; ---------------------------------------------------------------
          bra       install     ; installe le programme
                                ; comme résident
; ---------------------------------------------------------------
def_key:  dc.l      DEF_KEY     ; touche de définition
max_defs: dc.w      MAX_DEFS    ; nbre max de touches à
                                ; enregistrer
max_keys: dc.w      MAX_KEYS    ; nbre max de touches à
                                ; définir
; ---------------------------------------------------------------
new_trap:
; ---------------------------------------------------------------
          btst      #5,(a7)     ; mode système ?
          bne.s     sys_sp      ; oui
          move.l    usp,a0      ; non, prendre user stack pointer
          bra.s     check_bconin
sys_sp:
          lea       6(a7),a0    ; mode système, prendre le
                                ; system stack ptr
check_bconin:
          cmp.l     #$20002,(a0) ; dev=CON:, fonction=bconin
          bne.s     check_bconstat      ; fonction BIOS bconin

          bsr       new_bconin
          bra.s     rte
check_bconstat:
          cmp.l     #$10002,(a0) ; dev=CON:, fonction=bconstat
          bne       old_trap    ; ni bconin, ni bconstat
          bsr       new_bconstat
rte:
          rte
; ---------------------------------------------------------------
new_bconstat:         ; nouveau BCONSTAT
; ---------------------------------------------------------------
          move.l    nxt_char,d0 ; avant tout, traîte le
                                ; caractère en attente
          bne       char_avail  ; il y en a effectivement un
          tst.b     f_play      ; en train de re-jouer
                                ; une définition ?
          beq.s     get_stat    ; non, donc on prend le
                                ; status de CON:

          move.l    def_begin,a0
          cmp.l     def_end,a0  ; touches en attente
                                ; d'être retournées ?
          bne       char_avail  ; oui, la définition n'est
                                ; pas terminée

          clr.b     f_play      ; non, fin du playback
get_stat:
          bsr       old_constat
          tst       d0
          beq       stat_rts    ; pas de caractère disponible
          bsr       old_conin   ; prendre la caractère en attente
          move.l    d0,nxt_char ; et le sauver

          tst.b     f_record    ; en train d'enregistrer ?
          beq       tst_start_rec       ; non

          cmp.l     def_key,d1  ; fin de l'enregistrement ?
          beq       end_record  ; oui

          move.l    keys_free,a0 ; enregistre le caractère
                                ; dans la définition
          addq      #4,-2(a0)   ; incrémente la longueur de
                                ; la définition
          move.l    defs_free,a0
          move.l    d0,(a0)+
          move.l    a0,defs_free ; maj de la fin de la
                                ; définition

          cmp.l     defs_end,a0 ; fin de la table ?
          bne       char_avail  ; non

          clr.b     f_record
          signal_fault          ; signale la fin de la table
          bra       char_avail

end_record:
          signal_end
          clr.b     f_record    ; fin de l'enregistrement
          clr.l     nxt_char    ; supprime le caractère
                                ; de fin de définition
          move.l    current_def,d1
          bsr       find_key
          tst       d2          ; teste la longueur de la
                                ; définition
          bne.s     get_stat    ; pas vide
          bsr       kill_key    ; vide, donc on la supprime
          bra       get_stat

tst_start_rec:
          cmp.l     def_key,d1  ; début de définition ?
          beq.s     start_def   ; oui, alors on la démarre

          bsr       find_key    ; teste si c'est le
                                ; caractère de re-définit.
          bne       char_avail  ; non, on retourne

          st.b      f_play      ; on démarre le playback
          move.l    a0,def_begin
          add       d2,a0
          move.l    a0,def_end
          clr.l     nxt_char
          bra       char_avail

; Note Du Claviste (NDC): quand reverrai-je mon petit liré?

start_def:
          signal_start
          clr.l     nxt_char
          bsr       old_conin   ; on prend le carctère à définir
          cmp.l     def_key,d1
          beq       new_def_key ; on redéfinit le caractère
                                ; de définition

          move.l    d1,current_def      ; pour détruire la
                                ; définition si aucun caractère
                                ; n'est enregistré
          bsr       find_key    ; déjà une def pour ce caractère?
          bne       add_def     ; non, donc on l'ajoute
          bsr       kill_key    ; oui, alors on la supprime
add_def:
          move.l    keys_free,a0
          cmp.l     keys_end,a0 ; assez de place pour une
                                ; nouvelle déf. ?
          bne.s     def_char    ; oui

          signal_fault          ; non
          bra       get_stat

def_char:
          move.l    d1,(a0)+
          clr.w     (a0)+
          move.l    a0,keys_free
          st.b      f_record    ; on marque qu'on est en
                                ; train d'enregistrer
          bra       get_stat

new_def_key:
          bsr       old_conin
          move.l    d1,def_key  ; on sauve la nouvelle
                                ; touche de définition
          signal_end
          bra       get_stat

char_avail:
          moveq     #-1,d0      ; on marque qu'il y a un car.
                                ; de disponible
stat_rts:
          rts
; ---------------------------------------------------------------
new_bconin:           ; nouveau BCONIN
; ---------------------------------------------------------------
          bsr       new_bconstat
          tst       d0
          beq.s     new_bconin  ; attente d'un caractère

          move.l    nxt_char,d0
          beq.s     get_from_def

          clr.l     nxt_char
          bra.s     drop_kbs

get_from_def:
          move.l    def_begin,a0
          move.l    (a0)+,d0
          move.l    a0,def_begin

drop_kbs:
          btst      #3,(conterm).w      ; l'appelant souhaite
                                ; le status du clavier ?
          bne.s     ret_key     ; oui

          and.l     #$00ffffff,d0       ; non, on vire le
                                ; statut du clavier
ret_key:
          rts
          subttl    SUPPRIME LA DÉFINITION DE LA TOUCHE
; ---------------------------------------------------------------
kill_key: ; supprime la touche et sa définition
; ---------------------------------------------------------------
; entrée : A0.L -> début de la définition
;          A1.L -> code de touche
;          D2.L =  longueur de la définition
```

suite en page 27

UTILISER GEM ?
3615 STMAG: *GEM !

suite de la page 26

```
        sub.l   d2,defs_free
        sub.l   #6,keys_free
kk_nxt1:
        move.b  6(a1),(a1)+
        cmp.l   keys_free,a1
        blt.s   kk_nxt1
kk_nxt2:
        move.b  0(a0,d2),(a0)+
        cmp.l   defs_free,a0
        blt.s   kk_nxt2

        rts

        subttl  cherche un code de touche
; ---------------------------------------------------------------
find_key:  ; trouve une touche et sa définition
; ---------------------------------------------------------------
; entrée : D1.L = touche (avec 0 ou 2 shifts)
        ; sortie : Z si trouvée
;       A0  -> début de la définition
;       A1  -> code de touche
;       D2.W =  longueur de la définition

        move.l  keys_begin,a1
        move.l  keys_free,d2
        sub.l   a1,d2
        divu    #6,d2           ; nombre de touches à tester
        move.l  defs_begin,a0
        dbra    d2,fk_cmp
        tst     d2              ; remise à 0 du flag Z
fk_rts:
        rts
fk_nxt:
        add.w   (a1)+,a0        ; ajoute la longueur de la def
fk_cmp:
        cmp.l   (a1)+,d1
        dbeq    d2,fk_nxt

        bne.s   fk_rts          ; pas trouvé

        move.w  (a1),d2
        subq    #4,a1           ; pointe sur le début
                                ; de la définition
        cmp     d0,d0           ; positionne le flag Z
        rts

        subttl  PREND LE STATUT DU CLAVIER
; ---------------------------------------------------------------
old_constat:            ; prend le statut du clavier
; ---------------------------------------------------------------
; sortie : D0 = -1 si un caractère est disponible
        tst.b   f_record
        beq.s   start_stat

        physbase
        move.l  d0,a0
        add     #32000-80-82,a0
        move.b  flip,d0
        ror.b   #1,d0
        move.b  d0,(a0)         ; amuse l'utilisateur
                                ; pendant la définition
        move.b  d0,flip
start_stat:
        move.l  old_trap+2,a0           ; sauve l'adresse de
                                        ; la nouvelle routine
        setexec (a0),#45        ; restaure l'ancien vecteur
        move.l  d0,current_trap         ; garde le vecteur
                                        ; actuel du trap 13

        bconstat #2

        move.l  d0,-(a7)        ; sauve le registre
        move.l  current_trap,a0
        setexec (a0),#45        ; restaure le vecteur courant
        move.l  (a7)+,d0        ; restaure le registre

        rts

        subttl  LIT 1 CARACTERE AU CLAVIER
; ---------------------------------------------------------------
old_conin: ; prend le caractère au clavier
; ---------------------------------------------------------------
; sortie : D0 = caractère en entrée
;          D1 = D0 avec 2 touche shift keys si au
;               au moins une est pressée
; utilise : D2
        move.l  old_trap+2,a0           ; sauve l'adresse de la
                                        ; routine de trap
        setexec (a0),#45        ; restaure l'ancien vecteur
        move.l  d0,current_trap         ; garde le vecteur
                                        ; trap 12 courant
        move.b  (conterm).w,old_conterm
        bset    #3,(conterm).w          ; afin que le statut du
                                        ; clavier soit dans
                                        ; l'octet de poids fort
        bconin  #2

        move.b  old_conterm,(conterm).w ; restaure conterm

        move.l  d0,-(a7)        ; sauve le caractère
        move.l  current_trap,a0
        setexec (a0),#45        ; restaure le vecteur courant
        move.l  (a7)+,d0        ; restaure le caractère
proc_char:
        move.l  #$03000000,d2           ; masque pour les
                                        ; touches shift
        move.l  d0,d1
        and.l   d2,d1
        beq.s   oc_rts                  ; aucune touche
                                        ; shift appuyée
        move.l  d2,d1
oc_rts:
        or.l    d0,d1
        and.l   #$0fffffff,d1           ; enlève le flag
                                        ; caps lock
        rts

        subttl  DATA
;
;;      section DATA
;
        even
current_trap:   dc.l    $54321  ; vecteur trap 13
                                ; courant quand le programme
                                ; est appelé.
old_trap:  jmp  $12345.l        ; conserve l'ancien vecteur
trap 13
current_def:    dc.l    0       ; caractère en train
                                ; d'être défini
nxt_char:  dc.l 0               ; caractère en attente
f_record:  dc.b 0               ; <> 0 pendant
                                ; l'enregistrement
f_play:         dc.b    0       ; <> 0 pendant le
                                ; playback
flip:           dc.b    3       ; utilisé pour dire à
                                ; l'utilisateur qu'on est en
                                ; train de définir
snd_start: dc.b 0,$30           ; son de début de définition
           dc.b 1,0
           dc.b 2,0
           dc.b 3,0
           dc.b 4,0
           dc.b 5,0
           dc.b 6,0
           dc.b 7,$fe
           dc.b 8,$10
           dc.b 9,0
           dc.b 10,0
           dc.b 11,0
           dc.b 12,$10
           dc.b 13,9
           dc.b $ff,0
snd_end:                        ; son de fin de définition
           dc.b 0,$28
           dc.b 1,0
           dc.b 2,0
           dc.b 3,0
           dc.b 4,0
           dc.b 5,0
           dc.b 6,0
           dc.b 7,$fe
           dc.b 8,$10
           dc.b 9,0
           dc.b 10,0
           dc.b 11,0
           dc.b 12,$10
           dc.b 13,9
           dc.b $ff,0
snd_fault:                      ; son de table pleine
           dc.b 0,$0
           dc.b 1,1
           dc.b 2,0
           dc.b 3,0
           dc.b 4,0
           dc.b 5,0
           dc.b 6,0
           dc.b 7,$fe
           dc.b 8,$10
           dc.b 9,0
           dc.b 10,0
           dc.b 11,0
           dc.b 12,$20
           dc.b 13,9
           dc.b $ff,0
; ---------------------------------------------------------------
;       section BSS
; ---------------------------------------------------------------
        even

defs_begin:     ds.l    1       ; pointe sur le 1er
                                ; caractère de la 1ère def
defs_end:       ds.l    1       ; pointe après la zone de def
defs_free:      ds.l    1       ; pointe sur la 1ère place
                                ; libre pour l'enregistrement
def_begin:      ds.l    1       ; pointe sur la def à re-jouer
def_end:        ds.l    1       ; pointe sur la fin de 7 def
keys_begin:     ds.l    1       ; pointe sur la 1ère
                                ; touche définie
keys_end:       ds.l    1       ; pointe après la zone des
touches définies
keys_free:      ds.l    1       ; pointe sur la 1ère place
                                ; libre pour les touches à définir
old_conterm:    ds.b    1       ; ancienne valeur de conterm
; ---------------------------------------------------------------
        even
; début des définitions, suivies par les touches définies
reservation:

        subttl  INSTALLATION
        page
;       section TEXT
; ---------------------------------------------------------------
install:                ; début du programme, installe la partie
résidente
; ---------------------------------------------------------------
        lea     stack,a7        ; assigne le stack
        printline logo          ; affiche le logo
        supexec set_conterm
        bsr     dsp_def_key     ; affiche la touche de def
        bne.s                   no_error

        bconin  #2              ; attend un caractère
        term_2  #-1000          ; termine avec statut d'erreur
no_error:
        setexec new_trap,#45 ; change le vecteur TRAP 13
                ; pour qu'il pointe sur ce programme
        move.l  d0,old_trap+2 ; sauve l'ancien vecteur TRAP

        printline msg_q_keys
        move    max_keys,d0
        bsr     dsp_decimal
        printline msg_q_defs
        move    max_defs,d0
        bsr     dsp_decimal
        printline crlf

        lea     reservation,a0          ; calcule le pointeur
                                        ; de définitions
        move.l  a0,defs_begin
        move.l  a0,defs_free
        move    max_defs,d0
        mulu    #4,d0
        add     d0,a0
        move.l  a0,defs_end

        move.l  a0,keys_begin           ; calcule le pointeur
de touches
        move.l  a0,keys_free
        move    max_keys,d0
        mulu    #6,d0
        add     d0,a0
        move.l  a0,keys_end

        sub.l   #start_prg-256,a0
        keepproc #0,a0 ; fin du programme, qui reste résident

; ---------------------------------------------------------------
; positionne le bit 3 de la variable BIOS "conterm"
set_conterm
; ---------------------------------------------------------------
        bset            #3,conterm
        rts
; ---------------------------------------------------------------
dsp_def_key:            ; affiche la touche de définition
; ---------------------------------------------------------------
; sortie : flag Z si erreur

        move.b  def_key,d3
        btst    #0,d3
        beq.s   tst_ctrl
        printline msg_shift
tst_ctrl:
        btst    #2,d3
        beq.s   tst_alt
        printline msg_ctrl
tst_alt:
        btst    #3,d3
        beq.s   dsp_char
        printline msg_alt
dsp_char:
        tst.b   def_key+2
        bne.s   bad_def_key

        keytbl  #-1,#-1,#-1
        move.l  d0,a0
        move.l  (a0),a0
        move.b  def_key+1,d3 ; lecture du scan code
        bmi.s   bad_def_key ; doit être < 128

        ext     d3
        move.b  (a0,d3),d0
        beq.s   not_printable   ; caractère non imprimable

        cmp.b   #' ',d0
        ble.s   not_printable

        cmp.b   #$7f,d0
        beq.s   not_printable

        bconout #2,d0
        and     #%11111011,ccr          ; efface le flag Z
        rts     ; pour indiquer aucune erreur
not_printable:
        lea     tab_char,a0
        lsl     #2,d3           ; pour accéder au mot long
        move.l  (a0,d3.w),a0
        printline (a0)
        and     #%11111011,ccr          ; efface le flag Z
        rts     ; pour indiquer aucune erreur
bad_def_key:
        printline msg_err
        or      #%100,ccr       ; positionne le flag Z pour
        rts     ; indiquer une erreur

; ---------------------------------------------------------------
dsp_decimal:            ; affiche D0.L en décimal
; ---------------------------------------------------------------
        and.l   #$ffff,d0
        divu    #1000,d0
        move.l  d0,-(sp)        ; sauve le reste
        moveq   #2,d1           ; affiche les milliers
        lea     decimal_nr,a0
```

suite en page 28

KEY_DEFS

68000

suite de la page 27

```
         bsr         i2s
         move.b      #'.',(a0)+
         move.l      (sp)+,d0
         swap        d0
         moveq       #3,d1
         bsr         i2s
                                ; enlève les zéros du début
         moveq       #4,d0
         lea         decimal_nr,a0
dd_nxt:
         cmp.b       #'0',(a0)
         beq.s       clr_0

         cmp.b       #'.',(a0)
         bne.s       print_dec
clr_0:
         move.b      #' ',(a0)+
         dbra        d0,dd_nxt

print_dec:
         printline decimal_nr
         rts
; ---------------------------------------------------------
i2s      ; convertit un entier en une chaîne de caractères
; ---------------------------------------------------------
; entrée : d0, entier
;             d1, nombre de chiffres
;             a0, adresse de destination de la chaîne
; sortie : a0, pointe après le dernier caractère
         move        d1,-(sp)
         and.l       #$ffff,d0
         bra.s       i2s2
i2s1:
         divu        #10,d0
         swap        d0
         add         #'0',d0
         move.b      d0,(a0,d1)
         clr         d0
         swap        d0
i2s2:    dbra        d1,i2s1

         add         (sp)+,a0
         rts

;---------------------------------------------------------
;        données pour la partie non résidente
;---------------------------------------------------------

logo:    dc.b        13,10,"--------------------"
         dc.b        13,10,"Définition de touches.  (8-May-91)"
         dc.b        13,10,"--------------------"
         dc.b        13,10,"(c) Michel DESSAINTES"
         dc.b        13,10
         dc.b        13,10,"Touche de définition :"
         dc.b        13,10,"      ",0
crlf:    dc.b        13,10,0

msg_err:    dc.b     13,10,7,"Mauvaise touche de définition,"
            dc.b     13,10,"Installation NON faite.",0
msg_q_keys: dc.b     13,10,"Nbre max de définitions : ",0
msg_q_defs: dc.b     13,10,"Nbre max à enregistrer  : ",0
decimal_nr: dc.b     '12.345',0  ; nombre décimal à afficher
msg_alt:    dc.b     "Alt + ",0
msg_ctrl:   dc.b     "Ctrl + ",0
msg_shift:  dc.b     "Shift + ",0

         even
tab_char:             ; pointeurs sur les descriptifs de touche
         dc.l        sc_00,sc_01,sc_00,sc_00         ; 00
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 08
         dc.l        sc_00,sc_00,sc_0e,sc_0f
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 10
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 18
         dc.l        sc_1c,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 20
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 28
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 30
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_39,sc_00,sc_3b         ; 38
         dc.l        sc_3c,sc_3d,sc_3e,sc_3f
         dc.l        sc_40,sc_41,sc_42,sc_43         ; 40
         dc.l        sc_44,sc_00,sc_00,sc_47
         dc.l        sc_48,sc_00,sc_4a,sc_4b         ; 48
         dc.l        sc_00,sc_4d,sc_4e,sc_00
         dc.l        sc_50,sc_00,sc_52,sc_53         ; 50
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 58
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_61,sc_62,sc_63         ; 60
         dc.l        sc_64,sc_65,sc_66,sc_67
         dc.l        sc_68,sc_69,sc_6a,sc_6b         ; 68
         dc.l        sc_6c,sc_6d,sc_6e,sc_6f
         dc.l        sc_70,sc_71,sc_72,sc_00         ; 70
         dc.l        sc_00,sc_00,sc_00,sc_00
         dc.l        sc_00,sc_00,sc_00,sc_00         ; 78
         dc.l        sc_00,sc_00,sc_00,sc_00

sc_00:   dc.b        "Touche_inconnue",0
sc_01:   dc.b        "ESC",0
sc_0e:   dc.b        "BACKSPACE",0
sc_0f:   dc.b        "TAB",0
sc_1c:   dc.b        "RETURN",0
sc_39:   dc.b        "ESPACE",0
sc_3b:   dc.b        "F1",0
sc_3c:   dc.b        "F2",0
sc_3d:   dc.b        "F3",0
sc_3e:   dc.b        "F4",0
sc_3f:   dc.b        "F5",0
sc_40:   dc.b        "F6",0
sc_41:   dc.b        "F7",0
sc_42:   dc.b        "F8",0
sc_43:   dc.b        "F9",0
sc_44:   dc.b        "F10",0
sc_47:   dc.b        "CLR/HOME",0
sc_48:   dc.b        "Curseur Haut",0
sc_4a:   dc.b        "KP-",0    ; KP = clavier numérique
sc_4b:   dc.b        "Curseur Gauche",0
sc_4d:   dc.b        "Curseur Droite",0
sc_4e:   dc.b        "KP-+",0
sc_50:   dc.b        "Curseur Bas",0
sc_52:   dc.b        "INSERT",0
sc_53:   dc.b        "DELETE",0
sc_61:   dc.b        "UNDO",0
sc_62:   dc.b        "HELP",0
sc_63:   dc.b        "KP-(",0
sc_64:   dc.b        "KP-)",0
sc_65:   dc.b        "KP-/",0
sc_66:   dc.b        "KP-*",0
sc_67:   dc.b        "KP-7",0
sc_68:   dc.b        "KP-8",0
sc_69:   dc.b        "KP-9",0
sc_6a:   dc.b        "KP-4",0
sc_6b:   dc.b        "KP-5",0
sc_6c:   dc.b        "KP-6",0
sc_6d:   dc.b        "KP-1",0
sc_6e:   dc.b        "KP-2",0
sc_6f:   dc.b        "KP-3",0
sc_70:   dc.b        "KP-0",0
sc_71:   dc.b        "KP-.",0
sc_72:   dc.b        "ENTER",0

; ---------------------------------------------------------
         ds.l        20               ; stack pour la partie
                                      ; non résidente
stack:
         end
```

DEADKEYS

suite de la page 25

```
         dc.b        "'a",0,$A0        ; a accent aigu
         dc.b        "'i",0,$A1        ; i accent aigu
         dc.b        "'o",0,$A2        ; o accent aigu
         dc.b        "'u",0,$A3        ; u accent aigu
         dc.b        "n~",0,$A4        ; n tilde
         dc.b        "N~",0,$A5        ; N tilde
         dc.b        "_a",0,$A6        ; a souligné
         dc.b        "_o",0,$A7        ; o souligné
         dc.b        "??",0,$A8        ; ? inversé
         dc.b        "(-",0,$A9        ; coin haut gauche
         dc.b        ")-",0,$AA        ; coin haut droit (= not)
         dc.b        "12",0,$AB        ; fraction un demi
         dc.b        "14",0,$AC        ; fraction un quart
         dc.b        "!!",0,$AD        ; ! inversé
         dc.b        "<<",0,$AE        ; guillemet de gauche
         dc.b        ">>",0,$AF        ; guillemet de droite

         dc.b        "a~",0,$B0        ; a tilde
         dc.b        "o~",0,$B1        ; o tilde
         dc.b        "/O",0,$B2        ; O barré
         dc.b        "/o",0,$B3        ; o barré
         dc.b        "eo",0,$B4        ; ligatureo e
         dc.b        "EO",0,$B5        ; ligature O E
         dc.b        "A`",0,$B6        ; A accent grave
         dc.b        "A~",0,$B7        ; A tilde
         dc.b        "O~",0,$B8        ; O tilde
         dc.b        " ¨",0,$B9,' "',0,$B9  ; tréma (seul)
         dc.b        " '",0,$BA        ; accent aigu
         dc.b        "-|",0,$BB        ; croix
         dc.b        "!p",0,$BC        ; paragraph
         dc.b        "CO",0,$BD        ; copyright
         dc.b        "OR",0,$BE        ; registered
         dc.b        "MT",0,$BF        ; trade mard
         dc.b        "ij",0,$C0        ; i j
         dc.b        "IJ",0,$C1        ; I J
; et maintenant, les caractères hébreux
; (pourquoi ne pas avoir mis les correspondants IBM ?)
; NDC: parce que Tramiel est feuj, pardi!
         dc.b        " X",0,$C2        ; aleph
         dc.b        "-c",0,$C3
         dc.b        "-d",0,$C4
         dc.b        "-e",0,$C5
         dc.b        "-f",0,$C6
         dc.b        "-g",0,$C7
         dc.b        "-h",0,$C8
         dc.b        "-i",0,$C9
         dc.b        "-j",0,$CA
         dc.b        "-k",0,$CB
         dc.b        "-l",0,$CC
         dc.b        "-m",0,$CD
         dc.b        "-n",0,$CE
         dc.b        "-o",0,$CF
         dc.b        "-p",0,$D0
         dc.b        "-q",0,$D1
         dc.b        "-r",0,$D2
         dc.b        "-s",0,$D3
         dc.b        "-t",0,$D4
         dc.b        "-u",0,$D5
         dc.b        "-v",0,$D6
         dc.b        "-w",0,$D7
         dc.b        "-x",0,$D8
         dc.b        "-y",0,$D9
         dc.b        "-z",0,$DA
         dc.b        "-a",0,$DB
         dc.b        "-b",0,$DC  ; fin des caractères hébreux
         dc.b        "os",0,$DD  ; section
         dc.b        " L",0,$DE  ; lambda majuscule
         dc.b        "oo",0,$DF  ; infini
; maintenant, quelques caractères grecs
         dc.b        " a",0,$E0  ; alpha
         dc.b        " b",0,$E1  ; beta
         dc.b        " G",0,$E2  ; gamma majuscule
         dc.b        " p",0,$E3  ; pi
         dc.b        " S",0,$E4  ; sigma majuscule
         dc.b        " s",0,$E5  ; sigma
         dc.b        " n",0,$E6  ; nu
         dc.b        " t",0,$E7  ; tau
         dc.b        "O|",0,$E8  ; phi majuscule
         dc.b        " T",0,$E9  ; tau majuscule
         dc.b        " O",0,$EA,"-O",0,$EA  ; theta majuscule
         dc.b        " d",0,$EB  ; delta
         dc.b        "(o",0,$EC  ; circular integral
         dc.b        " f",0,$ED,"o|",0,$ED  ; phi (?)
         dc.b        " E",0,$EE  ; appartient
         dc.b        " P",0,$EF  ; pi majuscule

         dc.b        "-=",0,$F0  ; équivalent
         dc.b        "+-",0,$F1  ; plus ou moins
         dc.b        ">_",0,$F2  ; plus grand ou égal
         dc.b        "<_",0,$F3  ; plus petit ou égal
         dc.b        "(.",0,$F4  ; intégrale (partie supérieure)
         dc.b        ").",0,$F5  ; intégrale (partie inférieure)
         dc.b        "-:",0,$F6  ; divisé par
         dc.b        "~~",0,$F7  ; à peu près égal
         dc.b        " o",0,$F8  ; degré
         dc.b        "*o",0,$F9  ; point supérieur
         dc.b        ".^",0,$FA  ; point milieu
         dc.b        "-v",0,$FB  ; racine carrée
         dc.b        "^n",0,$FC  ; indice supérieur n
         dc.b        "2^",0,$FD  ; indice supérieur 2
         dc.b        "3^",0,$FE  ; indice supérieur 3
         dc.b        " ",0,$FF   ; ligne suppérieure

ct_end:                 ; fin de la table de conversion
;---------------------------------------------------------
         even

current_trap:        dc.l      $54321 ; vecteur trap 13
                               ; courant quand le programme
                               ; est appelé
old_trap: jmp        $12345.l  ; garde l'ancienne valeur
                               ; du TRAP 13

end_resident:                  ; fin de la partie résidente

;---------------------------------------------------------
; début du programme, installe la partie résidente
install:
;---------------------------------------------------------

         lea         stack,a7      ; assigne le stack
         printline   logo          ; affiche le logo
         setexec     new_trap,#45 ; change le vecteur TRAP 13 vector
                     ; pour qu'il pointe dans le programme
         move.l                 d0,old_trap+2    ; sauve
                                ; l'ancienne valeur du TRAP 13
         keepproc    #0,#end_resident-start_prg+256
                                ; fin du programme en laissant
                                ; une partie résidente

;---------------------------------------------------------
;        donnée pour la partie non résidente
;---------------------------------------------------------
logo:    dc.b        13,10,"--------------"
         dc.b        13,10,"Touches Mortes.  (8-May-91)"
         dc.b        13,10,"--------------"
         dc.b        13,10,"(c) Michel DESSAINTES"
         dc.b        13,10,0
;---------------------------------------------------------
         ds.l        20     ; stack pour la partie résidente
stack:
         end
```

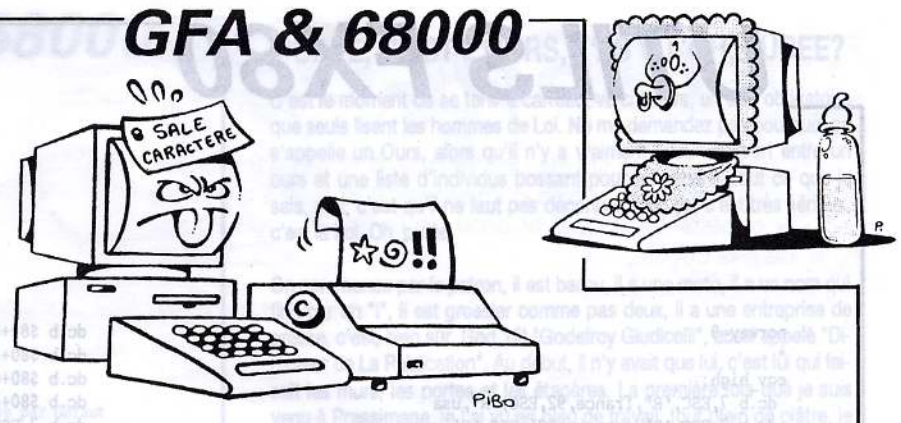
STMAGICIEL se fait des ennemis chez les feignasses, les fumistes, les fainéants, les nullards, les mutilés de guerre, les types pressés, les miros et les femmes enceintes.

L'accès au 3615 STMAG, le serveur où tous ces listings sont téléchargeables en deux temps trois mouvements, l'accès à ce serveur leur est interdit.

Ils n'ont qu'à être comme tout le monde, et par charité chrétienne, on ne vous parlera même pas des Bretons. C'est pas l'envie qui nous manque...

UTILS FX80

GFA & 68000

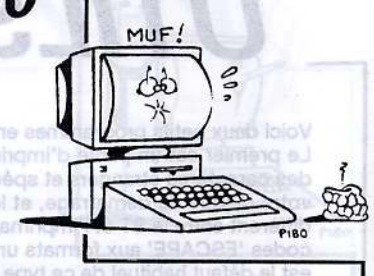


Voici deux petits programmes en assembleur, destinés aux possesseurs d'imprimantes 9 aiguilles compatibles EPSON.
Le premier est un pilote d'imprimante (nommé aussi "driver" par les anglomanes) : il permet d'imprimer la quasi-totalité des caractères étrangers et spéciaux du ST. Une fois assemblé avec l'extension PRG, placez-le dans le dossier AUTO de votre disque de démarrage, et le tour est joué : le programme interceptera et traitera tous les caractères dont les codes diffèrent entre le ST et l'imprimante. Il comporte aussi des garde-fous (les fous en question étant représentés par certains codes 'ESCAPE' aux formats un peu "tordus"...) pour éviter les perturbations de l'impression en mode graphique, ce qui est le défaut habituel de ce type d'utilitaire.
Le second est destiné à être intégré dans vos propres programmes, et a pour but de télécharger un jeu de caractère au format EPSON FX vers la mémoire de votre imprimante. Il suffit de passer sur la pile l'adresse en mémoire de la fonte à envoyer, cette dernière devant être au format GEM. Un exemple simplifié et minimaliste de chargement d'une fonte GEM en basic GFA 3 est joint. Le programme assemblé peut être intégré dans une commande INLINE du GFA 3. Attention : la résolution des imprimantes 9 aiguilles étant ce qu'elle est, les fontes utilisables sont limitées à 9 lignes en hauteur, et 11 pixels en largeur. Cela permet tout de même d'employer, par exemple, les fontes DEGAS converties en demi-hauteur avec le programme FONTCONV.PRG. Il sera souvent utile de les retravailler avec un éditeur de fontes GEM, afin de supprimer les points contigÜs horizontalement, qui ne seraient pas imprimés correctement.

Emmanuel TALMY

UTILITAIRE
Toutes Résolutions

```
*****************************************
* Un driver pour les imprimantes EPSON FX *
* Mai 1991              par NCC1701 *
*****************************************
*
        SECTION TEXT
*
* Détermine la configuration actuelle imprimante
        move    #-1,-(a7)
        move    #33,-(a7)
        trap    #14
        addq.w  #4,a7
* 'device' reçoit le numéro du port imprimante
        move    #0,device
        btst    #4,d0
        beq.s   zou
        move    #1,device
zou:
* Revectorise le BIOS pour gérer la fonction BConout
        pea     drv_bios(pc)
        move    #45,-(a7)
        move    #5,-(a7)
        trap    #13
        addq.w  #8,a7
        move.l  d0,stnd_bios+2
* Revectorise le XBIOS pour gérer la fonction SetPrt
        pea     drv_xbios(pc)
        move    #46,-(a7)
        move    #5,-(a7)
        trap    #13
        addq.w  #8,a7
        move.l  d0,stnd_xbios+2
* Initialise les variables et drapeaux internes
        sf      esc_typ1
        sf      esc_typ2
        sf      esc_typ3
        sf      esc_typ4
        sf      esc_flag
        clr     esc_decount
* Affiche le message d'installation
        pea     msg(pc)
        move    #9,-(a7)
        trap    #1
        addq.w  #6,a7
* Quitte le programme en le laissant résident
        move.l  4(a7),a5            Adresse basepage
        move.l  #256,d0             Longueur basepage
        add.l   12(a5),d0           + longueur Text
        add.l   20(a5),d0           + longueur Data
        add.l   28(a5),d0           + longueur Bss
        clr     -(a7)
        move.l  d0,-(a7)            Taille à réserver
        move    #$31,-(a7)          TERMRES
        trap    #1
        ILLEGAL
*************************
* Nouveau vecteur XBIOS *
*************************
drv_xbios:
* Pointeur sur les paramètres du TRAP #14 dans A0
        lea     6(a7),a0
        btst    #5,(a7)
        bne.s   .drv01
        move.l  usp,a0
.drv01:
* SI c'est un appel à SetPrt...
        cmpi    #33,(a0)+
        bne.s   stnd_xbios
* ... pour changer la configuration imprimante,
        move    (a0),d0
        bmi.s   stnd_xbios
* mémorise dans 'device' le nouveau port imprimante
        move    #0,device           Centronics (parallèle)
        btst    #4,d0
        beq.s   .drv02
        move    #1,device           ou RS232 (série)
.drv02:
* et continue avec le XBIOS standard
stnd_xbios jmp  $12345678
************************
* Nouveau vecteur BIOS *
************************
drv_bios:
* Pointeur sur les paramètres du TRAP #13 dans A0
        lea     6(a7),a0
        btst    #5,(a7)
        bne.s   .drv01
        move.l  usp,a0
.drv01:
* SI c'est un appel à BConout (fonction no 3)...
        moveq   #3,d0
        swap    d0
* ... vers le port de l'imprimante,
        move    device(pc),d0
        cmp.l   (a0)+,d0
        bne     stnd_bios
*
        tst.b   esc_flag            SI c'est un code ESCAPE,
        beq.s   drv20
drv10   move    (a0),d0             Code ESCAPE à traiter
        andi    #$7F,d0             varie de 0 à 127
* Détermine si c'est un code à nombre de paramètres variable
        lea     tordus(pc),a0       Liste de ces codes
        moveq   #n_typ1-1,d1        Type 1=téléchargement fonte
.drv10  cmp.b   (a0)+,d0
        dbeq    d1,.drv10
        seq     esc_typ1            Marque type 1 reconnu
        moveq   #n_typ2-1,d1        Type 2=données graphiques
.drv11  cmp.b   (a0)+,d0
        dbeq    d1,.drv11
        seq     esc_typ2            Marque type 2 reconnu
        moveq   #n_typ3-1,d1        Type 3=tabulations
.drv12  cmp.b   (a0)+,d0
        dbeq    d1,.drv12
        seq     esc_typ3            Marque type 3 reconnu
        moveq   #n_typ4-1,d1        Type 4=longueur de page
.drv13  cmp.b   (a0)+,d0
        dbeq    d1,.drv13
        seq     esc_typ4            Marque type 4 reconnu
* Détermine le nombre de paramètres fixe du code
        lea     fix_parm(pc),a0
        move.b  0(a0,d0),d0
        move    d0,esc_decount
        sf      esc_flag            Marque code ESCAPE reconnu
        sf      tordu_flag
        clr.b   tordu_val           Compteur de tabulations à 0
        bra     stnd_bios           Envoi du caractère...
*
drv20   tst     esc_decount         S'il reste des paramètres,
        beq.s   drv30
        subq    #1,esc_decount      -1 paramètre
        bra     stnd_bios           Envoi du car tel quel...
*
drv30   move.b  esc_typ1,d0         SI typ1 ou type2 en cours,
        or.b    esc_typ2,d0         (2 octets nécessaires)
        beq.s   drv40
        tst.b   tordu_flag          SI c'est le 1er octet,
        bne.s   .drv31
        move.b  1(a0),tordu_val     mémorise cet octet
        st      tordu_flag          marque premier octet lu
        bra.s   .drv39              et l'envoie à l'imprimante.
.drv31  move.b  1(a0),d0            SI c'est le 2nd octet,
        sf      tordu_flag          marque second octet lu
        tst.b   esc_typ1            SI gestion du type 1,
        beq.s   .drv32
        sub.b   tordu_val,d0        nombre d'octets de données=
        ext     d0                  12*(n2-n1+1)
        addq    #1,d0
        mulu    #12,d0
        sf      esc_typ1            Marque type 1 terminé
        bra.s   .drv33
.drv32  asl     #8,d0               SI gestion du type 2,
        move.b  tordu_val,d0        nombre d'octets de données=
        sf      esc_typ2  n1+256*n2
.drv33  move    d0,esc_decount      Initialise le décomptage
.drv39  bra     stnd_bios           Envoi du caractère...
*
drv40   tst.b   esc_typ3            SI type 3 en cours,
        beq.s   drv50
        addq.b  #1,tordu_val        +1 tabulation
        cmpi.b  #16,tordu_val       SI 16 tabulations reçues,
        beq.s   .drv41
        move.b  1(a0),d0            ou code NUL,
        bne.s   .drv49
.drv41  sf      esc_typ3            Marque type 3 terminé
.drv49  bra     stnd_bios           Envoi du caractère...
*
drv50   tst.b   esc_typ4            SI type 4 en cours,
        beq.s   drv60
        clr     esc_decount
        move.b  1(a0),d0            si code NUL,
        bne.s   .drv59
        addq    #1,esc_decount      +1 caractère à filtrer
.drv59  sf      esc_typ4            Marque type 4 terminé
        bra     stnd_bios           Envoi du caractère
*
drv60:
*
drv100  cmpi.b  #27,1(a0)           SI caractère ESCAPE,
        seq     esc_flag            marque ESCAPE en cours
*
        move.b  1(a0),d0            Code ASCII à envoyer
        andi    #$FF,d0             0 à 255
        cmpi.b  #32,d0              SI code<32,
        bcc.s   drv110
        lea     eqv_low(pc),a0      Table des codes ASCII<32
        bra.s   .low03              Recherche dans la table
.low01  cmp.b   d0,d1               SI caractère trouvé,
        beq.s   seq_send            on envoie sa séquence
.low02  move.b  tordu_val,d1        SINON, caractère suivant
        andi    #$7F,d1
        add.w   d1,a0
.low03  move.b  (a0)+,d1            Code caractère courant,
        move.b  (a0)+,tordu_val     Longueur de la séquence
        bne.s   .low01              SI fin de table,
        bra     stnd_bios           Envoi direct du code ASCII
*
drv110  cmpi.b  #128,d0             SI code>127,
        bcs     stnd_bios
        subi    #128,d0
        lea     eqv_high(pc),a0     Table des codes ASCII>127
        bra.s   .hig02              Recherche dans la table
.hig01  move.b  (a0)+,d1            Saute la séquence courante
        andi    #$7F,d1
        add.w   d1,a0
.hig02  dbra    d0,.hig01
        move.b  (a0)+,tordu_val     Longueur de la séquence
*
* Envoi d'une séquence équivalente à l'imprimante
seq_send:
        tst.b   tordu_val           SI longueur séquence<0,
        bpl.s   .seq01              -> séquence graphique
        move.l  a0,-(a7)
        lea     seq_graph(pc),a0    Imprimante mode graphique
        moveq   #n_graph-1,d0
        bsr.s   seq_output
        move.l  (a7)+,a0
.seq01  move.b  tordu_val(pc),d0    Envoi de la séquence
        andi    #$7F,d0             équivalente
        subq    #1,d0
        bsr.s   seq_output
        tst.b   tordu_val           SI séquence graphique,
        bpl.s   .seq02
        lea     seq_nulnul(pc),a0   2 octets nuls à la fin
        moveq   #n_nulnul-1,d0
        bsr.s   seq_output
.seq02  rte                         Retour après le TRAP
#13
*
seq_output:
        move.b  (a0)+,d1
        movem.l d0/a0,-(a7)         Sauvegarde pointeurs
        move    d1,-(a7)
        move    device,-(a7)        Port imprimante
        move    #3,-(a7)            Code BConout
        pea     .seq03(pc)          Simule un TRAP #13
        move    sr,-(a7)            avec retour en .drv103
        bra     stnd_bios
.seq03  addq.w  #6,a7
        movem.l (a7)+,d0/a0         Restaure les pointeurs
        dbra    d0,seq_output       Boucle sur la séquence...
        rts
*
stnd_bios jmp   $12345678 Saut au BIOS standard
*
        SECTION DATA
*
* Liste des codes ESCAPE de format spécial
*
tordus:
typ1    dc.b    '&'                 Type 1 = Fonte EPSON
n_typ1=*-typ1
typ2    dc.b    '*','K','L','Y','Z' Type 2 = Graphismes
n_typ2=*-typ2
typ3    dc.b    'B','D','b'         Type 3 = Tabulations
n_typ3=*-typ3
typ4    dc.b    'C'                 Type 4 = Longueur page
n_typ4=*-typ4
*
* Nombre de paramètres fixe pour chaque ESC (0 à 127)
*
fix_parm:
        dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        dc.b    0,1,0,0,0,2,1,0,0,0,1,0,0,1,0,1
        dc.b    0,0,0,1,0,0,0,0,0,0,3,0,0,0,0,0
        dc.b    0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0
        dc.b    0,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0
        dc.b    0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0
        dc.b    1,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0
*
* Séquences pour l'impression en mode graphique
seq_graph       dc.b    ESC,'L',12,0,0,0
n_graph=*-seq_graph
seq_nulnul      dc.b    0,0
n_nulnul=*-seq_nulnul
*
* Tables de conversion des caractères spéciaux
*
* I) Codes ASCII 0 à 31, sauf codes de commandes EPSON
eqv_low:
 dc.b 1,$80+8,$18,$38,$6F,$C1,$C1,$6F,$38,$18
 dc.b 2,$80+8,$18,$1C,$F6,$83,$83,$F6,$1C,$18
 dc.b 3,$80+8,$3C,$24,$24,$E7,$C3,$66,$3C,$18
 dc.b 4,$80+8,$18,$3C,$66,$C3,$E7,$24,$24,$3C
 dc.b 5,$80+8,$7C,$38,$92,$C6,$C6,$92,$38,$7C
 dc.b 6,$80+8,$FE,$FE,$FE,$FC,$F8,$F2,$E6,$CE
 dc.b 7,$80+8,$EE,$C6,$92,$38,$38,$92,$C6,$EE
 dc.b 26,$80+8,$00,$0C,$2E,$2A,$2A,$3E,$1C,$00
 dc.b 28,$80+8,$00,$00,$00,$3F,$7A,$E3,$E3,$E1
 dc.b 29,$80+8,$E0,$E1,$E2,$C3,$62,$3F,$00,$00
 dc.b 30,$80+8,$00,$00,$07,$83,$67,$3C,$5C,$EA
 dc.b 31,$80+8,$2A,$2A,$52,$2C,$70,$80,$00,$00
 dc.b 0,0
*
* II) Codes ASCII 128 à 255
*
BS=8
ESC=27
usa=0
france=1
germany=2
england=3
italy=6
spain=7
japan=8
```



TT & MEGA STE : *TT SUR 3615 STMAG

```
norway=9
*
eqv_high:
 dc.b 7,ESC,'R',france,92,ESC,'R',usa
 dc.b 7,ESC,'R',germany,125,ESC,'R',usa
 dc.b 7,ESC,'R',france,123,ESC,'R',usa
 dc.b 3,'a',BS,'^'
 dc.b 7,ESC,'R',germany,123,ESC,'R',usa
 dc.b 7,ESC,'R',france,64,ESC,'R',usa
 dc.b 7,ESC,'R',norway,125,ESC,'R',usa
 dc.b 7,ESC,'R',france,92,ESC,'R',usa
 dc.b 3,'e',BS,'^'
 dc.b 9,'e',BS,ESC,'R',france,126,ESC,'R',usa
 dc.b 7,ESC,'R',france,125,ESC,'R',usa
 dc.b 9,'i',BS,ESC,'R',france,126,ESC,'R',usa
 dc.b 3,'i',BS,'^'
 dc.b 7,ESC,'R',italy,126,ESC,'R',usa
 dc.b 7,ESC,'R',germany,91,ESC,'R',usa
 dc.b 7,ESC,'R',norway,93,ESC,'R',usa
 dc.b 7,ESC,'R',norway,64,ESC,'R',usa
 dc.b 7,ESC,'R',norway,123,ESC,'R',usa
 dc.b 7,ESC,'R',norway,91,ESC,'R',usa
 dc.b 3,'o',BS,'^'
 dc.b 7,ESC,'R',germany,124,ESC,'R',usa
 dc.b 7,ESC,'R',italy,124,ESC,'R',usa
 dc.b 3,'u',BS,'^'
 dc.b 7,ESC,'R',france,124,ESC,'R',usa
 dc.b 9,'y',BS,ESC,'R',france,126,ESC,'R',usa
 dc.b 7,ESC,'R',germany,92,ESC,'R',usa
 dc.b 7,ESC,'R',germany,93,ESC,'R',usa
 dc.b 3,'c',BS,'|'
 dc.b 7,ESC,'R',england,'#',ESC,'R',usa
 dc.b 7,ESC,'R',japan,92,ESC,'R',usa
 dc.b 7,ESC,'R',germany,126,ESC,'R',usa
 dc.b $80+8,$00,$22,$7E,$FC,$A0,$A0,$80,$00
 dc.b 3,'a',BS,39
 dc.b 3,'i',BS,39
 dc.b 3,'o',BS,39
 dc.b 3,'u',BS,39
 dc.b 7,ESC,'R',spain,124,ESC,'R',usa
 dc.b 7,ESC,'R',spain,92,ESC,'R',usa
 dc.b 3,'a',BS,95
 dc.b 3,'o',BS,95
 dc.b 7,ESC,'R',spain,93,ESC,'R',usa
 dc.b $80+8,$00,$00,$1E,$1E,$10,$10,$10,$00
 dc.b $80+8,$00,$10,$10,$10,$1E,$1E,$00,$00
 dc.b $80+8,$E6,$EC,$18,$30,$69,$D3,$9F,$0D
 dc.b $80+8,$E6,$EC,$18,$36,$6A,$DF,$9F,$02
 dc.b $80+8,$00,$00,$00,$5F,$5F,$00,$00,$00
 dc.b $80+8,$10,$38,$6C,$D6,$BA,$6C,$C6,$82
 dc.b $80+8,$82,$C6,$6C,$BA,$D6,$6C,$38,$10
 dc.b 3,'a',BS,'~'
 dc.b 3,'o',BS,'~'
 dc.b 7,ESC,'R',norway,92,ESC,'R',usa
 dc.b 7,ESC,'R',norway,124,ESC,'R',usa
 dc.b $80+8,$1C,$3E,$22,$3E,$3E,$2A,$3A,$18
 dc.b $80+8,$7C,$FE,$82,$FE,$FE,$92,$92,$82
 dc.b $80+8,$00,$07,$8F,$DA,$5A,$0F,$07,$00
 dc.b $80+8,$00,$47,$8F,$DA,$5A,$8F,$07,$00
 dc.b $80+8,$00,$5E,$BF,$E1,$61,$BF,$1E,$00
 dc.b 7,ESC,'R',france,126,ESC,'R',usa
 dc.b 1,39
 dc.b $80+8,$00,$00,$20,$7C,$20,$00,$00,$00
 dc.b $80+8,$70,$F8,$88,$88,$FF,$00,$FF,$00
 dc.b $80+8,$7E,$C3,$BD,$BD,$A5,$A5,$C3,$7E
 dc.b $80+8,$7E,$C3,$BD,$A9,$AD,$B5,$C3,$7E
 dc.b $80+8,$80,$F8,$80,$F8,$60,$30,$60,$F8
 dc.b $80+8,$24,$BC,$BC,$05,$01,$BF,$BE,$00
 dc.b $80+8,$84,$FC,$FC,$85,$01,$FF,$FE,$00
 dc.b $80+8,$00,$6C,$7C,$30,$18,$7C,$6C,$00
 dc.b $80+8,$00,$44,$44,$44,$7C,$7C,$04,$00
 dc.b $80+8,$00,$00,$04,$4C,$58,$7C,$7C,$00
 dc.b $80+8,$00,$40,$40,$40,$7C,$7C,$40,$00
 dc.b $80+8,$00,$5C,$5C,$40,$40,$7C,$3C,$00
 dc.b $80+8,$00,$00,$00,$40,$7C,$7C,$00,$00
 dc.b $80+8,$00,$00,$00,$40,$60,$7C,$5C,$00
 dc.b $80+8,$00,$40,$7C,$7C,$40,$7C,$7C,$00
 dc.b $80+8,$00,$FC,$FC,$04,$44,$7C,$7C,$00
 dc.b $80+8,$00,$00,$40,$40,$70,$70,$00,$00
 dc.b $80+8,$00,$00,$44,$44,$44,$7C,$7C,$00
 dc.b $80+8,$00,$C0,$C0,$40,$44,$7C,$7C,$00
 dc.b $80+8,$00,$5C,$7C,$20,$64,$7C,$3C,$00
 dc.b $80+8,$00,$00,$04,$44,$7C,$7C,$00,$00
 dc.b $80+8,$00,$00,$78,$7C,$44,$7C,$78,$00
 dc.b $80+8,$00,$04,$7C,$7C,$04,$7C,$7C,$00
 dc.b $80+8,$00,$74,$74,$54,$44,$7C,$7C,$00
 dc.b $80+8,$00,$64,$74,$14,$1C,$7C,$6C,$00
 dc.b $80+8,$00,$00,$5E,$5E,$40,$7C,$78,$00
 dc.b $80+8,$00,$40,$40,$40,$7C,$3C,$00,$00
 dc.b $80+8,$7C,$7C,$04,$7C,$04,$7C,$7C,$00
 dc.b $80+8,$04,$7C,$7C,$40,$7C,$7C,$00,$00
 dc.b $80+8,$00,$00,$00,$40,$7E,$7E,$00,$00
 dc.b $80+8,$00,$00,$40,$40,$40,$7E,$7E,$00
 dc.b $80+8,$40,$7C,$7C,$44,$44,$7C,$7C,$00
 dc.b $80+8,$00,$70,$70,$50,$40,$7E,$7E,$00
 dc.b $80+8,$00,$00,$60,$70,$1E,$7E,$60,$00
 dc.b 7,ESC,'R',france,93,ESC,'R',usa
 dc.b $80+8,$0C,$18,$30,$60,$30,$18,$0C,$00
 dc.b $80+8,$78,$CC,$CC,$70,$38,$CC,$CC,$78
 dc.b $80+8,$1C,$3E,$22,$36,$1C,$36,$22,$00
 dc.b $80+8,$00,$3F,$7F,$D2,$92,$FE,$6C,$00
 dc.b $80+8,$41,$7F,$7F,$41,$41,$60,$70,$00
 dc.b $80+8,$20,$3F,$3E,$20,$3F,$3E,$20,$00
 dc.b $80+8,$82,$C6,$EE,$BA,$92,$C6,$C6,$00
 dc.b $80+8,$00,$1C,$3E,$62,$7E,$5C,$40,$00
 dc.b $80+8,$01,$3F,$3E,$02,$3E,$3E,$02,$02
 dc.b $80+8,$00,$20,$20,$3F,$3E,$20,$20,$00
 dc.b $80+8,$00,$18,$BD,$E7,$E7,$BD,$18,$00
 dc.b $80+8,$00,$3C,$7E,$52,$52,$7E,$3C,$00
 dc.b $80+8,$00,$3A,$7E,$40,$40,$7E,$3A,$00
 dc.b $80+8,$1C,$3E,$66,$F2,$BE,$DC,$40,$00
 dc.b $80+8,$00,$19,$27,$7E,$E4,$98,$00,$00
 dc.b $80+8,$1C,$3E,$22,$7F,$22,$3E,$1C,$00
 dc.b $80+8,$00,$7C,$FE,$D6,$92,$92,$92,$00
 dc.b $80+8,$00,$7E,$FE,$80,$80,$FE,$7E,$00
 dc.b $80+8,$00,$54,$54,$54,$54,$54,$54,$00
 dc.b $80+8,$00,$22,$22,$FA,$FA,$22,$22,$00
 dc.b $80+8,$00,$02,$8A,$DA,$72,$22,$02,$00
 dc.b $80+8,$00,$02,$22,$72,$DA,$8A,$02,$00
 dc.b $80+8,$00,$00,$00,$3F,$7F,$40,$70,$30
 dc.b $80+8,$0C,$0E,$02,$FE,$FC,$00,$00,$00
 dc.b $80+8,$00,$10,$10,$D6,$D6,$10,$10,$00
 dc.b $80+8,$00,$24,$48,$48,$24,$24,$48,$00
 dc.b $80+8,$00,$40,$E0,$A0,$E0,$40,$00,$00
 dc.b $80+8,$00,$40,$E0,$E0,$E0,$40,$00,$00
 dc.b $80+8,$00,$00,$00,$0C,$0C,$00,$00,$00
 dc.b $80+8,$08,$0C,$06,$1E,$38,$20,$20,$20
 dc.b $80+8,$00,$78,$F8,$80,$F8,$78,$00,$00
 dc.b $80+8,$00,$48,$D8,$B8,$E8,$48,$00,$00
 dc.b $80+8,$00,$88,$A8,$A8,$F8,$50,$00,$00
 dc.b $80+8,$40,$40,$40,$40,$40,$40,$40,$00
*
* Message de copyright au chargement en AUTO
msg     dc.b    13,10
        dc.b    27,'p DRIVER pour EPSON FX80'
        dc.b    ' V1.1 ',27,'q',13,10
        dc.b    189,'Mai 1991        par'
        dc.b    ' NCC1701',13,10
        dc.b    0
*
        SECTION BSS
        EVEN
device          ds.w    1
esc_decount     ds.w    1
esc_flag        ds.b    1
esc_typ1        ds.b    1
esc_typ2        ds.b    1
esc_typ3        ds.b    1
esc_typ4        ds.b    1
tordu_flag      ds.b    1
tordu_val       ds.b    1
*
        END
```

```
' ***********************************************************************
' * Exemple de téléchargement de fontes GEM vers une EPSON FX           *
' * Cet accessoire permet d'envoyer une fonte sur une imprimante EPSON *
' * Pour réactiver la fonte interne de l'imprimante, choisir 'ANNULER' *
' * dans le sélecteur de fichiers.                                      *
' ***********************************************************************
'
' Options de compilation pour les routines ASM
$C+
$m5000
' INLINE contenant la routine ASM de téléchargement
INLINE fnt_fx80%,432
' Enregistrement de l'accesoire
ap_id&=APPL_INIT()
menu_id&=MENU_REGISTER(ap_id&,". Fontes EPSON FX ")
IF menu_id&<0 THEN
  ~EVNT_TIMER(-1)
ENDIF
' Initialise les variables disques
font_drive&=GEMDOS(25)
font_path$=DIR$(0)
' Adresse du tampon de messages GEM
mesag_buffer%=MENU(-2)
' Initialise la gestion des erreurs
ON ERROR GOSUB handle_error
' Boucle principale de l'accessoire
DO
  ~EVNT_MESAG(mesag_buffer%)
  IF INT{mesag_buffer%}=40 THEN
    @select_font
  ENDIF
after_error:
LOOP
END
'
' ************************************
' * Choix d'une fonte GEM sur disque *
' ************************************
PROCEDURE select_font
  LOCAL font_addr%,font_err%
  font_path$=font_path$+"*.FNT"
  font_name$=""
  ~FSEL_INPUT(font_path$,font_name$,choix&)
  font_path$=LEFT$(font_path$,RINSTR(font_path$,"\"))
  IF choix&<>0 THEN                    ! SI choix 'CONFIRMER
    ' **** Chargement de la fonte choisie
    font_addr%=@font_load(font_path$+font_name$)
    ' *** SI chargement OK, envoi à l'imprimante et verdict
    IF font_addr%>0 THEN
      font_err%=C:fnt_fx80%(L:font_addr%)
      IF font_err%=1 THEN
        ALERT 3,"| Fonte de trop grande taille |",1," GLOP ",void&
      ELSE IF font_err%=0
        ALERT 3,"| Imprimante non prête |",1," GLOP ",void&
      ELSE IF font_err%=-1
        ' Active la police téléchargée en double impression
        LPRINT "%";CHR$(1);CHR$(0);"G";
        ALERT 0,"| La fonte '"+CHR{ADD(font_addr%,4)}+"' | est
téléchargée et activée.|",1,"CHOUETTE",void&
      ENDIF
      ~MFREE(font_addr%)               ! Libère la mémoire
    ENDIF
  ELSE                                 ! SI choix 'ANNULER',
    LPRINT "%";CHR$(0);CHR$(0);"H";    ! Réactive la police interne
    ALERT 0,"| La fonte standard de votre | imprimante est
réactivée. |",1," MERCI ",void&
  ENDIF
RETURN
' ************************************************************
' *                Chargement d'une fonte GEM                *
' *            (strict minimum pour l'application)           *
' ************************************************************
FUNCTION font_load(filename$)
  LOCAL addr%,table%,count&
  OPEN "I",#1,filename$
  addr%=MALLOC(LOF(#1))
  IF addr%>0 THEN
    BGET #1,addr%,LOF(#1)
    INT{ADD(addr%,36)}=ROR&(INT{ADD(addr%,36)},8)
    INT{ADD(addr%,38)}=ROR&(INT{ADD(addr%,38)},8)
    INT{ADD(addr%,52)}=ROR&(INT{ADD(addr%,52)},8)
    INT{ADD(addr%,66)}=ROR&(INT{ADD(addr%,66)},8)
    INT{ADD(addr%,72)}=ROR&(INT{ADD(addr%,72)},8)
    INT{ADD(addr%,74)}=ROR&(INT{ADD(addr%,74)},8)
    INT{ADD(addr%,76)}=ROR&(INT{ADD(addr%,76)},8)
    INT{ADD(addr%,78)}=ROR&(INT{ADD(addr%,78)},8)
    INT{ADD(addr%,80)}=ROR&(INT{ADD(addr%,80)},8)
    INT{ADD(addr%,82)}=ROR&(INT{ADD(addr%,82)},8)
    {ADD(addr%,72)}=ADD(addr%,SWAP({ADD(addr%,72)}))
    {ADD(addr%,76)}=ADD(addr%,SWAP({ADD(addr%,76)}))
    table%={ADD(addr%,72)}
    count&=MUL(2,SUCC(SUB(INT{ADD(addr%,38)},INT{ADD(addr%,36)})))
    WHILE count&>=0
      INT{ADD(table%,count&)}=ROR&(INT{ADD(table%,count&)},8)
      SUB count&,2
    WEND
    IF BTST(INT{ADD(addr%,66)},2) THEN
      table%={ADD(addr%,76)}
      count&=MUL(INT{ADD(addr%,80)},INT{ADD(addr%,82)})
      WHILE count&>0
        SUB count&,2
        INT{ADD(table%,count&)}=ROR&(INT{ADD(table%,count&)},8)
      WEND
    ENDIF
  ELSE
    ALERT 2,"| Mémoire insuffisante ! |",1," GNY ",count&
  ENDIF
  CLOSE #1
  RETURN addr%
ENDFUNC
' ********************************
' * Affichage de l'erreur survenue *
' ********************************
PROCEDURE handle_error
  LOCAL void&
  CLOSE
  ~MFREE(addr%)
  ~FRE(0)
  ALERT 2," | Erreur "+STR$(ERR),1," ET PAF ",void&
  ON ERROR GOSUB handle_error
  RESUME after_error
RETURN


' ******************************
' *  Crée l'Inline OUTPUT.INL  *
' *       à charger dans       *
' *       le listing GfA       *
' ******************************
'
yepa_ligne=0
OPEN "o",#1,"OUTPUT.INL"
DO
  INC yepa_ligne
  yepa_check=0
  FOR yepa_yepa=1 TO 16
    READ yepa_donnee$
    yepa_donnee=VAL("&h"+yepa_donnee$)
    EXIT IF yepa_donnee$="-1"
    ADD yepa_check,yepa_donnee
    OUT #1,yepa_donnee
  NEXT yepa_yepa
  EXIT IF yepa_donnee$="-1"
  READ yepa_ctrl$
  IF VAL("&h"+yepa_ctrl$)<>yepa_check
    PRINT "Erreur de DATA en ligne ";yepa_ligne;"."
    PRINT "Vous vous êtes trompé, c'est malin."
    END
  ENDIF
```

UTILS FX80

GFA & 68000

```
  PRINT "Ligne ";yepa_ligne;" exploitée...";CHR$(13);
LOOP
CLOSE #1
DATA 60,1A,0,0,1,5C,0,0,0,34,0,0,0,0,0,10B
DATA 0,0,0,0,0,0,0,0,0,0,0,0,70,1,20,6F,100
DATA 0,4,C,68,0,9,0,52,6E,0,1,1E,C,68,0,B,1DF
DATA 0,34,6E,0,1,14,3F,3C,FF,FF,3F,3C,0,21,4E,4E,468
DATA 58,4F,E8,40,2,40,0,1,41,FA,1,2E,30,80,3F,3A,4A5
DATA 1,28,3F,3C,0,8,4E,4D,58,4F,4A,40,67,0,0,EA,3C9
DATA 41,FA,1,44,70,4,61,0,0,E2,67,0,0,DC,4D,FA,5C1
DATA 1,A,22,6F,0,4,24,69,0,48,7E,0,41,FA,1,28,357
DATA 42,A0,42,A0,42,A0,30,7,C,40,0,20,6C,C,10,36,407
DATA 0,0,B0,69,0,26,6F,2,30,7,B0,69,0,24,6D,0,391
DATA 0,8E,B0,69,0,26,6E,0,0,86,D0,40,3C,32,0,2,441
DATA 30,32,0,0,9C,40,48,C0,80,FC,0,8,26,69,0,4C,4A5
DATA D6,C0,72,7,48,40,92,40,16,3C,0,80,34,29,0,52,4EA
DATA C,42,0,9,6D,2A,74,8,30,29,0,50,E7,40,49,F3,476
DATA 0,0,38,1,3A,6,60,14,9,14,67,8,42,3,D6,E9,37D
DATA 0,50,60,C,53,44,6C,4,52,4C,78,7,51,CD,FF,EA,5E7
DATA 86,46,10,C3,3A,6,60,22,70,7,28,4B,38,2,60,C,3F1
DATA 3,14,67,2,1,D0,D8,E9,0,50,53,40,51,CC,FF,F2,703
DATA 52,48,53,41,6A,4,72,7,52,4B,51,CD,FF,DC,41,FA,6E6
DATA 0,6A,70,B,61,0,0,14,67,0,0,E,52,47,C,47,2BB
DATA 0,7F,6F,0,FF,38,70,FF,4E,75,48,E7,60,E0,12,18,6F0
DATA 48,E7,80,80,3F,1,3F,3A,0,20,3F,3C,0,3,4E,4D,421
DATA 5C,4F,4A,80,4C,DF,1,1,57,C8,FF,E4,56,C0,48,80,782
DATA 48,C0,4C,DF,7,6,4E,75,0,0,85,8A,97,95,8D,F8,6C3
DATA 9C,AD,A8,A5,A4,99,D4,8F,86,87,DD,9E,92,91,B2,B3,A46
DATA B9,8E,99,9A,84,94,81,90,82,9D,88,0,0,0,FE,7C,7C4
DATA 38,10,0,0,0,0,1B,26,0,0,7F,0,0,0,0,0,108
DATA -1
```

```
        OUTPUT    .INL
*****************************************
*     Téléchargement d'une fonte GEM    *
* vers une imprimante compatible EPSON FX *
* Mai 1991              par NCC1701     *
*****************************************
*
* Paramètre attendu sur la pile : adresse de la fonte
                RSRESET
                rs.l      1
font_addr       rs.l      1

* Vérifie la taille de la fonte (9x11 maxi)
        moveq   #1,d0                   Code erreur 1
        move.l  4(a7),a0
        cmpi    #9,82(a0)               si fonte trop haute
        bgt     exit
        cmpi    #11,52(a0)              ou trop large
        bgt     exit

* Détermine le port de l'imprimante par SetPrt
        move    #-1,-(a7)
        move    #33,-(a7)
        trap    #14
        addq.w  #4,a7
        asr     #4,d0
        andi    #1,d0
        lea     device(pc),a0
        move    d0,(a0)                 Résultat dans device

* Teste si l'interface de l'imprimante est prête
        move    device(pc),-(a7)
        move    #8,-(a7)
        trap    #13
        addq.w  #4,a7
        tst     d0                      Code d'erreur=0
        beq     exit

* Envoi de l'en-tête de la séquence
        lea     font_esc(pc),a0
        moveq   #n_font_esc-1,d0
        bsr     seq_send
        beq     exit                    Code d'erreur=0

* Initialisations des principaux pointeurs
        lea     gem_flex(pc),a6         Table caractères étrangers
        move.l  4(a7),a1                Adresse de la fonte
        move.l  72(a1),a2               Table de décalage

        moveq   #0,d7                   Départ au code ASCII 0
char_loop:
        lea     buffer+12(pc),a0
        clr.l   -(a0)                   Mise à zéro du tampon
        clr.l   -(a0)                   graphique
        clr.l   -(a0)

* Vérifie la présence du caractère dans la fonte GEM
        move    d7,d0                   Code ASCII du caractère
        cmpi    #32,d0                  SI code<32,
        bge.s   .char01
        move.b  0(a6,d0),d0             correspondance fontes GEM
        cmp     38(a1),d0               Pas dans la fonte?
        ble.s   .char01
        move    d7,d0                   reprend le code original
.char01 cmp     36(a1),d0               SI le code ASCII n'est
        blt     send_char               pas défini dans la
        cmp     38(a1),d0               fonte, envoie du blanc
        bgt     send_char               (~ESPACE~)

* Détermine la position du caractère dans la fonte
        add     d0,d0                   Index décalage
        move    2(a2,d0),d6             Pos. départ car.suivant
        move    0(a2,d0),d0             Pos. départ car.courant
        sub     d0,d6                   Différence=largeur
        ext.l   d0
        divu    #8,d0                   8 bits par octet...

* Adresse du premier octet du caractère
        move.l  76(a1),a3               Table de données
        add.w   d0,a3                   +décalage

* Index du premier bit du caractère dans cet octet
        moveq   #7,d1
        swap    d0
        sub     d0,d1

* Calcul de la donnée proportionnelle
        move.b  #$80,d3                 Pas de jambage par défaut
        move    82(a1),d2               SI fonte de hauteur 9,
        cmpi    #9,d2
        blt.s   .pro09

        moveq   #8,d2                   8 lignes pour EPSON
        move    80(a1),d0               adresse dernière ligne
        asl     #3,d0                   du caractère
        lea     0(a3,d0),a4
        move    d1,d4                   Boucle de test
        move    d6,d5                   de la dernière ligne
        bra.s   .pro02
.pro01  btst    d4,(a4)                 SI pixel allumé,
        beq.s   .pro03
        clr.b   d3                      Active le jambage
        add.w   80(a1),a3               Saute la première ligne
        bra.s   .pro09
.pro03  subq    #1,d4                   SINON, pixel suivant
        bge.s   .pro02
        addq.w  #1,a4
        moveq   #7,d4
.pro02  dbra    d5,.pro01
.pro09  or      d6,d3                   Donnée proportinnelle
        move.b  d3,(a0)+                dans le tampon EPSON

* Mise au format EPSON FX, colonne par colonne
        move    d6,d5                   Nombre de colonnes
        bra.s   next_column             Boucle sur icelles
column_loop:
        moveq   #7,d0                   Colonne de gauche
        move.l  a3,a4                   Première ligne car.
        move    d2,d4                   Nombre de lignes
        bra.s   next_row                Boucle sur icelles
row_loop:
        btst    d1,(a4)                 SI pixel allumé dans
        beq.s   .row01                  la fonte,
        bset    d0,(a0)                 allume le pixel du tampon
.row01  add.w   80(a1),a4               Ligne suivante fonte
        subq    #1,d0                   Pixel suivant du tampon
next_row:
        dbra    d4,row_loop             -1 ligne
        addq.w  #1,a0                   Octet suivant du tampon
        subq    #1,d1                   Bit suivant de la fonte
        bpl.s   next_column             Si on change d'octet,
        moveq   #7,d1                   recommence au bit 7
        addq.w  #1,a3                   octet suivant de la fonte
next_column:
        dbra    d5,column_loop          -1 colonne

* Envoi du tampon représentant le caractère
send_char:
        lea     buffer(pc),a0
        moveq   #12-1,d0  12 octets
        bsr     seq_send
        beq     exit                    Code erreur 0 si Time-out
        addq    #1,d7                   caractère suivant,
        cmpi    #127,d7                 envoyer 128 caractères
        ble     char_loop
        moveq   #-1,d0                  Code -1, envoi OK
exit    rts

* Envoi d'une séquence d'octets à l'imprimante
* a0=adresse, d0=longueur-1
seq_send:
        movem.l d1-d2/a0-a2,-(a7)       Sauvegarde registres
.seq01  move.b  (a0)+,d1                Octet courant
        movem.l d0/a0,-(a7)             Sauvegarde pointeurs
        move    d1,-(a7)
        move    device(pc),-(a7)        Port imprimante
        move    #3,-(a7)                BConout
        trap    #13
        addq.w  #6,a7
        tst.l   d0                      Teste le time-out
        movem.l (a7)+,d0/a0             Restaure pointeurs
        dbeq    d0,.seq01               Boucle sur les octets
        sne     d0                      Renvoie d0=-1 si OK
        ext     d0                      d0=0 si Time out
        ext.l   d0
        movem.l (a7)+,d1-d2/a0-a2       Restaure registres
        rts
*
        SECTION   DATA
device          dc.w      0

* Liste des codes ASCII dans les fontes GEM
* des caractères EPSON 0 à 31 (étrangers)
gem_flex:
 dc.b   133,138,151,149,141,248,156,173
 dc.b   168,165,164,153,212,143,134,135
 dc.b   221,158,146,145,178,179,185,142
 dc.b   153,154,132,148,129,144,130,157

* Tampon graphique de 12 octets
buffer          ds.b      12

* Commande ESCAPE pour le téléchargement sur FX
font_esc  dc.b    27,'&',0,0,127
n_font_esc=*-font_esc
*
        END
```

Minitel STMAGICIEL:

3615 STMAG

PUREE, C'EST L'OURS, T'AS VU, OH, PUREE?

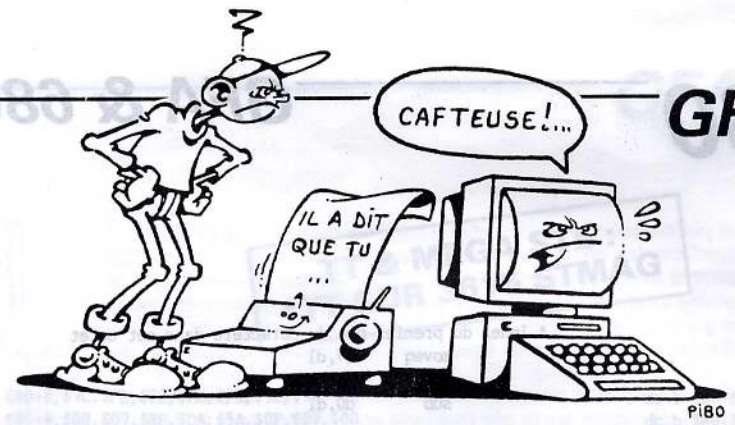
C'est le moment de se tenir à carreau, voici l'Ours, un truc obligatoire que seuls lisent les hommes de Loi. Ne me demandez pas pourquoi ça s'appelle un Ours, alors qu'il n'y a vraiment aucun rapport entre un ours et une liste d'individus bossant pour un canard, tout ce que je sais, moi, c'est qu'il ne faut pas déconner avec ça, c'est très sérieux, c'est la Loi. Oh, purée.

On commence par le patron, il est barbu, il a une moto, il a un nom qui finit par un "i", il est grossier comme pas deux, il a une entreprise de presse, c'est, bien sûr, God, dit "Godefroy Giudicelli", aussi appelé "Directeur de La Publication". Au début, il n'y avait que lui, c'est lui qui faisait les murs, les portes et les étagères. La première fois que je suis venu à Pressimage, je l'ai vu en bleu de travail, tout plein de plâtre, je pensais que c'était un ouvrier, je me suis drôlement retenu pour ne pas lui donner la bastonnade. On a changé de murs et de portes, mais on a gardé les étagères, ça lui aurait fait trop de peine, alors on les a encore, même si on sait pas quoi en faire, tu poses un magazine dessus et tout se casse la gueule, mais ça lui ferait trop de peine.
Ensuite, il y a le rédacteur en chef, il s'appelle Mic Dax, encore que ce soit pas un vrai rédacteur en chef dans la mesure où il n'y a pas grand chose à écrire dans un hors-série spécial listings, ça lui ressemble bien à ce fainéant de s'occuper d'un truc où y a pas à bosser, encore un coup pour se remplir les poches avec le blé du patron !
Après ça, il y a ceux-qui-ont-collaboré. Je vais pas tous vous les dénoncer, vous n'avez qu'à lire tous les noms en dessous de chaque présentation de programme, je les aurais tous cités si le canard était vendu 25 balles, mais pour 20 balles, faudrait voir à se calmer et à pas pousser. Alors, au début, il y a Jacques Caron, l'auteur d'Accload, Discpac et Sapristi. Il ne se sent plus depuis qu'il est le nouveau Rédac'chef de ST Magazine, il veut qu'on le vouvoie, qu'on aille lui acheter son Coca et qu'on balance ces "vieilles étagères de merde, non mais qu'est ce que c'est que cette pourriture, ça vous amuse de faire les décharges ?!" (je cite). Ensuite, il y a Oya Shivo, un grand dadais qui fait tout dans le désordre; il met ses pompes avant les chaussettes, devient célèbre avant de sortir le moindre petit soft, ou bien s'achète-et-conduit-une-belle-voiture-qui-coûte-je-sais-pas-combien-de-barres avant de passer le permis. Té, il est bon, Oya. Passons à Emmanuel Talmy, un vieux de la vieille sur ST, c'est l'auteur, entre autres, d'Intermusic et d'Interdigit, très sérieux, personne n'ose imaginer ce qui a dû lui arriver pour qu'il soit aussi pince-sans-rire, il ne parle plus que d'imprimantes et de trucs sous interruption, avec une voix très grave.
Rien à voir avec Fabrice Broche, l'auteur de... moui, passons... C'est un type pas triste du tout, on se téléphone très très souvent (comprenez "à chaque fois que son soft plante") et on se fend bien la poire, vous n'avez qu'à lire son article en fin de journal, et si ça ne vous fait pas hurler de rire, c'est que vous n'êtes qu'un imbécile, désolé. Non, même pas, je ne suis pas désolé, je n'ai pas à culpabiliser, vous êtes un imbécile, je n'y peux rien. Il y en a un qu'il ne faut pas oublier, c'est Sébastien Mougey, c'est l'auteur de File Hunter, de pas mal de logiciels pour des synthés dont je n'ai pas retenu le nom, et prochainement de Digital Impact, un soft suuuuuper bien, si vous ne l'achetez pas, c'est le même topo que pour tout à l'heure avec Fabrice. Plus sympa que Seb, ça n'existe pas, c'est une crème, et Agathe c'est sa copine. Pour finir, il y a Pibo, le super dessinateur, si tout va bien il y a un article quelque part dans le canard qui vous le présente. Voilà pour ces collaborateurs, les ouin-je-suis-pas-cité sont sympas aussi.

Ensuite, il y a la fabrication. Ca, normalement, c'est Michel qui s'en occupe. Toute une histoire, Michel, un ancien de la troupe, il a fait l'Indo et l'Afrique Noire. Quand on a fini ST Mag, on vient tous s'asseoir autour de lui pour qu'il nous raconte ses souvenirs et on s'endort tellement c'est nul, il nous sort des vannes minables mais on l'adore. Ce coup-ci, il était en vacances, il a juste montré en cinq minutes comment marchaient les 12 machines de l'atelier, il a promis que tout se passerait bien et il est parti en rigolant. La maquettiste, c'est Mireille, elle aussi en vacances, il a fallu se débrouiller, merci bien. Alors elle, c'est l'Espagne. Tu lui parles d'un truc qui n'a pas de rapport avec l'Espagne ou avec les trains, elle baille et tu as l'air con. Si tu veux te venger, tu l'obliges à parler espagnol, c'est à mourir de rire, on croirait Antoine en train d'imiter Godefroy parler anglais, on est tous pliés !
Antoine, c'est Antoine Harmel, le Directeur de la Publicité. Pour STMAGICIEL, il s'est pas foulé, vous avez remarqué, mais il n'était pas à son affaire, parce qu'il vient juste d'avoir son premier enfant. Il a eu la bonne idée de l'appeler Arthellaïs, c'est un prénom pratique à porter, c'est passe-partout, y a pas de problème. Antoine, quand je lui dis ça, il comprend bien que c'est que de la jalousie. La chef de pub, c'est Véronique Perrin, qui n'était pas non plus à son affaire, puisque son supérieur venait d'avoir son premier enfant. A deux, ils nous ont trouvé un seul annonceur, on n'est pas dans la mouise.
Bon, voilà, c'est fini. Les autres de l'équipe, ce sont des comptables, des commerciaux, des col-blancs, des conseillers, ils ne nous intéressent pas, ni vous ni moi, on va pouvoir passer à l'imprimeur. Ha, si, il y a Olivier Lepotvin, c'est le Chef des Ventes, c'est un type important, pour un canard ! Il a des lunettes.

L'imprimeur, c'est Paulo, qui travaille aux Imprimeries Mayennaises, à Mayenne, dans la Mayenne, au moins il ne s'est pas creusé la tête pour le nom. C'est toujours la femme à Paulo qui nous appelle, parce qu'avec tout le retard qu'on lui fait prendre à chaque fois, il est tellement furax qu'il n'est capable que de nous insulter, il a raison. Sinon, la photogravure couleur, vous vous en foutez mais c'est quand même Serge, de la Sima, qui nous l'a faite, et la photogravure noire, on l'a fait faire chez nous, on a un peu tout cassé et salopé mais on y est arrivé.
Comme d'habitude, il est interdit de recopier ou de traduire (surtout si c'est en breton !), même partiellement, nos textes, dessins, photos et linstingues sans notre autorisation. Ca ne veut pas dire que vous êtes obligés de nous appeler dès que vous voulez recopier un programme, ça s'adresse aux autres canards qui voudraient repiquer nos trucs à nous. Autrement dit, si vous bossez dans la presse, que ce soit pour le Washington Post ou pour le fanzine des Amis de la Philatélie en Aquitaine, et que vous voulez recopier un truc qu'on a fait, nous, eh bien vous nous appelez et on s'arrange, c'est tout.
Notre Numéro De Commission Paritaire, c'est celui de ST Magazine, puisque c'est son premier Hors-Série que vous tenez là, c'est le numéro, écoutez bien, 71845, je répète, 7, 1, 8, 4 et 5.
Ok, c'est fini, il n'y a plus rien à voir, et moi, je pars me dorer la peau au soleil, je vous embrasse très fort, ne faites pas de bêtises, et envoyez-nous des cartes postales, pas à l'adresse du Siège Social de Pressimage, la SARL de presse au capital de 2000 francs qui édite ST Magazine, pas au 210 rue du Faubourg Saint Martin 75010 PARIS, mais au 19 rue Hégésippe Moreau 75018 PARIS, c'est là que nous crèchons. Ciao, come back mit velocidad!

HARDCOPY



GFA BASIC

Choisissez à la souris une portion d'écran à imprimer, sélectionnez le mode d'impression (8 ou 24 aiguilles) et choisissez à quel endroit de la feuille vous voulez faire la copie.

Michel MOUSSIÉ

```
'                PROGRAMME D'ESSAI DES ROUTINES DE COPIE D'ECRAN
'                              "HARDCOPY_24"
'                        Michel Moussié - 13 juin 1991
ALERT 2,"L'imprimante est-elle|  opérationnelle?",0," SUITE |
INFOS ",ijk%
IF ijk%=2
  ALERT 0,"Programme HARDCOPY_24|   Michel Moussié|    13 juin
1991",0," SUITE ",ijk%
ENDIF
IF ijk%=1
  REPEAT
    GOSUB texte(0)          ! écran pour essai
    GOSUB fenetre           ! définition fenêtre à copier et mode
    SELECT cas%
    CASE 1                  ! hardcopy mode 8 aiguilles
      GOSUB texte(1)
      GOSUB copie_8
    CASE 2                  ! hardcopy mode 24 aiguilles
      GOSUB texte(1)
      GOSUB copie_24
    ENDSELECT
    ALERT 2,"Voulez-vous continuer|    ou abandonner",0," SUITE
|ABANDON",choix%
  UNTIL choix%=2
ENDIF
END
' ####################FENETRE##############################
PROCEDURE fenetre
  GRAPHMODE 3
  DO                        ! origine du cadre
    MOUSE x0%,y0%,k%
    EXIT IF k%=1
  LOOP
  x1%=x0%
  y1%=y0%
  REPEAT                    ! traçage du cadre
    DEFLINE 0,1,0,0
    BOX x0%,y0%,x1%,y1%
    MOUSE x1%,y1%,k%
    DEFLINE 1,1,0,0
    BOX x0%,y0%,x1%,y1%
  UNTIL k%=0
  DELAY 2
  DEFLINE 0,1,0,0           ! effacement du cadre
  BOX x0%,y0%,x1%,y1%
  ' y0%=ADD(y0%,24)         ! correction éventuelle pour
  ' y1%=ADD(y1%,24)         ! window 0
  GRAPHMODE 1
  CLS
  GOSUB message(1)          ! infos coordonnées cadre
  '                         VERIFICATION DES COORDONNEES CADRE
  IF x1%<=x0% OR y1%<=y0%
    GOSUB message(2)
  ELSE
    IF x1%>=640             ! débordement droit
      x1%=640
      dx1%=0
      octet_max%=80
    ELSE
      octet_max%=DIV(x1%,8)   ! dernier octet des lignes écran
      dx1%=MOD(x1%,8)       ! dépassement dans le dernier octet
    ENDIF
    '
    octet_min%=DIV(x0%,8)     ! premier octet des lignes écran
    dx0%=MOD(x0%,8)         ! dépassement dans le premier octets
    '
    GOSUB message(3)        ! choix 8 ou 24 aiguilles/abandon
    SELECT cas%
    CASE 1                  ! mode 8 aiguilles
      ligne_min%=DIV(y0%,8)   ! première ligne imprimante
      dy0%=MOD(y0%,8)       ! dépassement dans la 1ère ligne
      IF y1%>=400
        y1%=400
        dy1%=0
        ligne_max%=50
      ELSE
        ligne_max%=DIV(y1%,8) ! dernière ligne imprimante
        dy1%=MOD(y1%,8)  ! dépassement dans la dernière ligne
      ENDIF
    CASE 2                  ! mode 24 aiguilles
      ligne_min%=DIV(y0%,24)
      dy0%=MOD(y0%,24)
      IF y1%>=400
        y1%=400
        dy1%=16
        ligne_max%=17
      ELSE
        ligne_max%=DIV(y1%,24)
        dy1%=MOD(y1%,24)
      ENDIF
    ENDSELECT
    IF cas%=1 OR cas%=2
      GOSUB message(4)      ! acquisition du positionnement
      px%=px%*60/25.4       ! conversion en 1/60 pouce
      py%=py%*60/25.4
      '                     VERIFICATION DU POSITIONNEMENT X
      IF (x1%-x0%)/80+px%/60>8 AND cas%=1
        px%=480-(x1%-x0%)*3/4  ! correction dépassement 8 aiguilles
      ENDIF
      IF (x1%-x0%)/180+px%/60>8 AND cas%=2
        px%=480-(x1%-x0%)/3    ! correction dépassement 24 aiguilles
      ENDIF
      '
      r%=XBIOS(2)           ! adresse début mémoire écran
    ENDIF
  ENDIF
RETURN
' ####################COPIE_8##############################
PROCEDURE copie_8
  LPRINT CHR$(27);CHR$(79);     ! évitement perfo annulé
  LPRINT CHR$(27);"A";CHR$(1);  ! avance papier 1/60 pouce
  FOR i%=1 TO py%           ! offset Y sur papier
    LPRINT CHR$(10);
  NEXT i%
  '
  LPRINT CHR$(27);"3";CHR$(24);  ! avance papier 24/180 pouce
  LPRINT CHR$(27);"U";"0";       ! impression bidirectionnelle
  '                         CALCUL ET IMPRESSION DES LIGNES
  HIDEM
  bloc%=DIV((x1%-x0%),256)  ! nb de blocs de 256 col./ligne
  extra%=MOD((x1%-x0%),256) ! nb de colonnes en plus/ligne
  debut%=MOD(x0%,8)         ! début de ligne dans impr%()
  bloc_offset%=DIV(px%,256) ! nb de blocs d'offset de 256 col.
  x_offset%=MOD(px%,256)    ! nombre de col. d'offset en plus
  DIM impr%(x1%-x0%+24)     ! tableau des octets d'une ligne
  DIM lect|(7)
  FOR ligne%=ligne_min% TO ligne_max%  ! les lignes imprimante.
    a1%=ADD(r%,MUL(ligne%,640)) ! adr début de ligne:ligne%*8*80
    FOR octet%=octet_min% TO octet_max% ! octets des lignes écran
      a2%=ADD(a1%,octet%)           ! adr de l'octet courant.
      a3%=MUL(SUB(octet%,octet_min%),8)
      FOR i%=0 TO 7  ! les 8 octets de meme rang dans lignes
        lect|(i%)=BYTE{a2%+MUL(i%,80)} ! écran d'une ligne imprimante
        IF ligne%=ligne_min% AND i%<dy0% ! correction début de colonne
          lect|(i%)=0
        ENDIF
        IF ligne%=ligne_max% AND i%>dy1% ! correction fin de colonne
          lect|(i%)=0
        ENDIF
      NEXT i%
      FOR bit%=1 TO 8    ! calcul des octets imprimante
        a%=0
        FOR i%=0 TO 7
          ex%=2^SUB(7,i%)
          a%=ADD(a%,MUL(VAL(MID$(BIN$(lect|(i%),8),bit%,1)),ex%))
        NEXT i%
        impr%(ADD(a3%,bit%))=a%
      NEXT bit%
    NEXT octet%
    LPRINT CHR$(27);"$";CHR$(x_offset%);CHR$(bloc_offset%);
    ' offset X sur papier
    LPRINT CHR$(27);"*";CHR$(4);CHR$(extra%);CHR$(bloc%);
    ' mode CRT1
    FOR i%=debut%+1 TO debut%+x1%-x0%
      LPRINT CHR$(impr%(i%));
    NEXT i%
    LPRINT CHR$(10);     ! LF
  NEXT ligne%
  LPRINT
  ERASE impr%(),lect|()
  '
  GRAPHMODE 3
  DEFLINE 1,1,0,0
  ' y0%=SUB(y0%,24)       ! correction éventuelle pour
  ' y1%=SUB(y1%,24)       ! window 0
  BOX x0%,y0%,x1%,y1%
  GRAPHMODE 1
  SHOWM
RETURN
' ####################COPIE_24#############################
PROCEDURE copie_24
  LPRINT CHR$(27);CHR$(79);     ! évitement perfo annulé
  LPRINT CHR$(27);"A";CHR$(1);  ! avance papier 1/60 pouce
  FOR i%=1 TO py%           ! offset Y sur papier
    LPRINT CHR$(10);
  NEXT i%
  '
  LPRINT CHR$(27);"3";CHR$(24);  ! espace 24/180 pouce
  LPRINT CHR$(27);"U";"0";       ! impression bidirectionnelle
  '                         CALCUL ET IMPRESSION DES LIGNES
  HIDEM
  bloc%=DIV((x1%-x0%),256)  ! nb de blocs de 256 col./ligne
  extra%=MOD((x1%-x0%),256) ! nb de colonnes en plus/ligne
  debut%=MOD(x0%,8)         ! début de ligne dans impr%()
  bloc_offset%=DIV(px%,256) ! nb de blocs d'offset de 256 col.
  x_offset%=MOD(px%,256)    ! nombre de col. d'offset en plus
  DIM impr%(x1%-x0%+24,2)   ! tableau des octets d'une ligne
  DIM lect|(23)             ! tableau des octets de 8 col.
  FOR ligne%=ligne_min% TO ligne_max%
    a1%=ADD(r%,MUL(ligne%,1920))        ! adr début de ligne:
    FOR octet%=octet_min% TO octet_max% ! r%+(ligne%*24*80)
      a2%=ADD(a1%,octet%)    ! adr des octets de la ligne
      a3%=MUL(SUB(octet%,octet_min%),8) ! nb de 8xcol déjà traitées
      FOR i%=0 TO 23
        lect|(i%)=BYTE{a2%+MUL(i%,80)}
        IF ligne%=ligne_min% AND i%<dy0% ! correction déb de col
          lect|(i%)=0
        ENDIF
        IF ligne%=ligne_max% AND i%>dy1% ! correction fin de col
          lect|(i%)=0
        ENDIF
      NEXT i%
      FOR bit%=1 TO 8    ! 8 col. suivantes de la ligne
        a%=0
        b%=0
        c%=0
        FOR i%=0 TO 7    ! 8 bits des 3 octets de la col.
          ex%=2^SUB(7,i%)
          a%=ADD(a%,MUL(VAL(MID$(BIN$(lect|(i%),8),bit%,1)),ex%))
          b%=ADD(b%,MUL(VAL(MID$(BIN$(lect|(i%+8),8),bit%,1)),ex%))
          c%=ADD(c%,MUL(VAL(MID$(BIN$(lect|(i%+16),8),bit%,1)),ex%))
        NEXT i%
        impr%(ADD(a3%,bit%),0)=a%
        impr%(ADD(a3%,bit%),1)=b%
        impr%(ADD(a3%,bit%),2)=c%
      NEXT bit%
    NEXT octet%
    LPRINT CHR$(27);"$";CHR$(x_offset%);CHR$(bloc_offset%);
    ' offset sur papier
    LPRINT CHR$(27);"*";CHR$(39);CHR$(x1%-x0%);CHR$(bloc%);
    ' mode triple dens.
    FOR i%=debut%+1 TO debut%+x1%-x0%
      LPRINT CHR$(impr%(i%,0));CHR$(impr%(i%,1));CHR$(impr%(i%,2));
    NEXT i%
    LPRINT CHR$(10);                  ! LF
  NEXT ligne%
  LPRINT
  ERASE impr%(),lect|()
  '                         RETRACAGE DU CADRE
  GRAPHMODE 3
  DEFLINE 1,1,0,0
  ' y0%=SUB(y0%,24)       ! correction éventuelle pour
  ' y1%=SUB(y1%,24)       ! window 0
  BOX x0%,y0%,x1%,y1%
  GRAPHMODE 1
  '
  SHOWM
RETURN
' ##############################MESSAGE#######################
PROCEDURE message(numero%)
  SELECT numero%
  CASE 1
    PRINT AT(20,2);"Les coordonnées de la fenêtre choisie sont:"
    PRINT AT(30,3);"x0=";x0%
    PRINT AT(45,3);"y0=";y0%
    PRINT AT(30,4);"x1=";x1%
    PRINT AT(45,4);"y1=";y1%
  CASE 2
    ALERT 1,"Coordonnées de fenêtre|   erronnéees",0,"RETURN",ijk%
  CASE 3
    ALERT 2,"Voulez-vous une copie d'écran|     8 ou 24
aiguilles|        ou abandon",0," 8 | 24 |ABANDON",cas%
  CASE 4
    PRINT AT(15,6);"Quel positionnement sur le papier désirez-vous?"
    LOCATE 30,7
    INPUT "offset X en mm";px%
    LOCATE 30,8
    INPUT "offset Y en mm";py%
  ENDSELECT
RETURN
' ##############################TEXTE#########################
PROCEDURE texte(cas%)
  CLS
  PRINT
  PRINT "      ROUTINES DE COPIE D'ECRAN POUR IMPRIMANTE 24 AIGUILLES "
  PRINT "                  Michel Moussié - 13 juin 1991"
  PRINT
  PRINT "      Deux routines permettent de réaliser des copies
d'écran, l'une (COPIE_24)"
  PRINT " utilise le mode 24 aiguilles, l'autre (COPIE_8) le
mode 8 aiguilles. Une"
  PRINT " routine (FENETRE) permet de choisir une portion de
l'écran et de la copier"
  PRINT " à un endroit choisi de la feuille."
  PRINT
  PRINT "      Dans le mode 24 aiguilles 3 octets sont néces-
saires pour commander les"
  PRINT " 24 aiguilles de la tete d'impression tandis que dans
le mode 8 aiguilles"
  PRINT " un octet suffit pour commander une aiguille sur
trois de la tete. Il en"
  PRINT " résulte que en mode 24 aiguilles les dimensions de
la copie sont divisées"
  PRINT " par trois par rapport à celles en mode 8 aiguilles."
  PRINT
  PRINT "      Le corps du programme n'a pour but que de pouvoir
utiliser les routines."
  PRINT " La routine TEXTE génère cet écran d'essai qui donne
quelques explications."
  PRINT " Les messages des routines sont réunis dans la routi-
ne MESSAGE."
  PRINT
  PRINT "      La source HARDCOPY.GFA est documentée de façon à
pouvoir en suivre le"
  PRINT " fonctionnement tandis qu'une version compilée .PRG
permet de juger de la"
  PRINT " rapidité. Le programme a été testé avec une impri-
mante PANASONIC KX-P1124."
  PRINT CHR$(27);"p"
  IF cas%=0
    HTAB 5
    PRINT "GENERER UNE FENETRE A L'AIDE DE LA SOURIS ET DE SON
BOUTON DE GAUCHE"
  ENDIF
  IF cas%=1
    HTAB 25
    PRINT "ATTENDEZ, CALCUL EN COURS."
  ENDIF
  PRINT CHR$(27);"q";
RETURN
```

UTILITAIRE
Toutes Résolutions

Minitel STMAGICIEL:

3615 STMAG

Téléchargez les listings publiés dans ce numéro.

Protocole SAPRISTI.

GFA-PUNCHS

Des GfA-Punchs, ce sont des listings en GfA de 20 lignes grand maximum, on n'a pas le droit de les dépasser sinon ça compte pas.

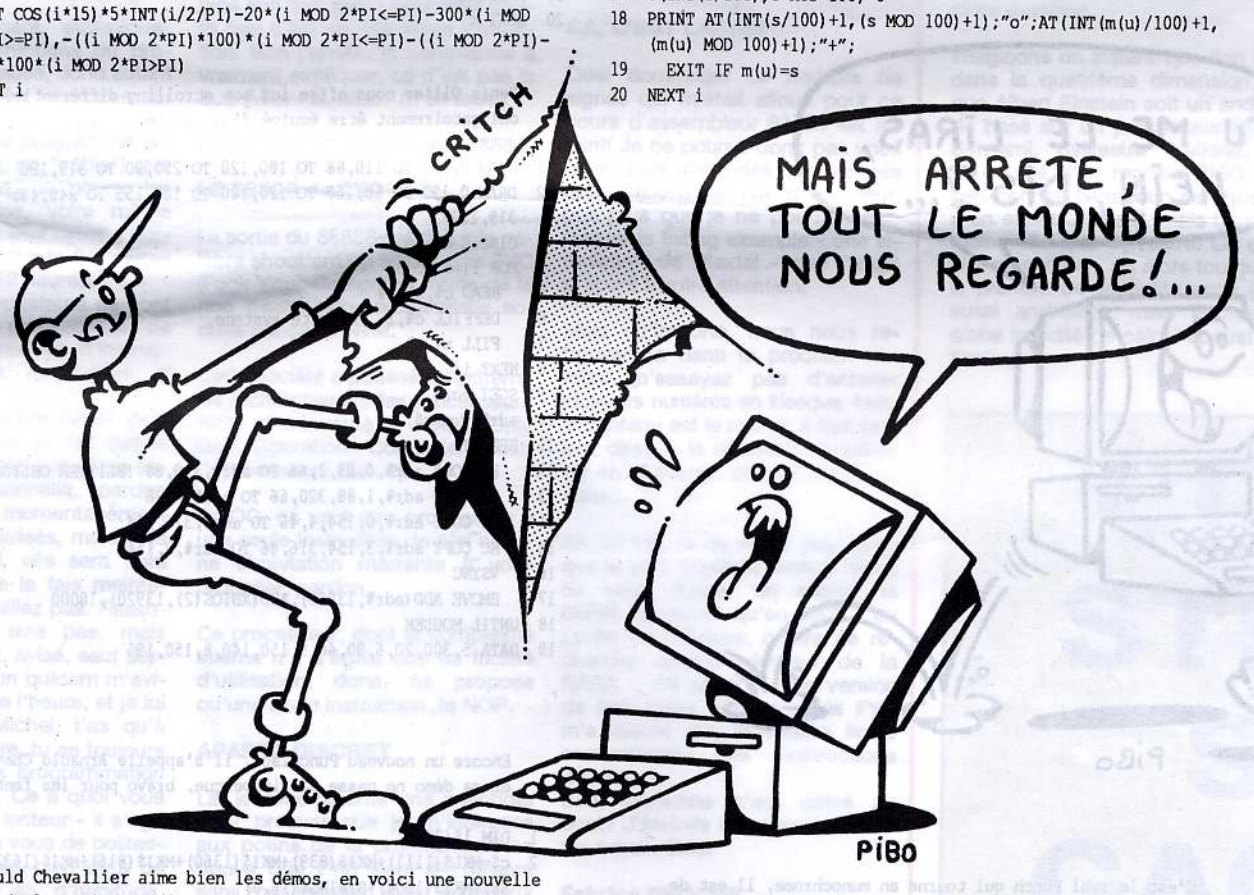
Tonton PUNCH

Voici un nouveau venu qui fait très fort, il s'agit de Flagnum, qui commence par une démo paramétrique.

```
1  GRAPHMODE 2
2  FOR i=1 TO 15
3  COLOR i
4  LINE 0,i,319,i
5  NEXT i
6  TEXT 80,11," THE PARAMETRIC DEMO"
7  GET 80,1,260,15,p$
8  a$=MKI$(1)+MKI$(16)+MKI$(1)+MKI$(0)+MKI$(1)+MKI$(112)+MKI$(32)+
   MKI$(248)+MKI$(80)+MKI$(508)+MKI$(168)+MKI$(1022)+MKI$(332)+
   MKI$(2047)+MKI$(658)+MKI$(4095)+MKI$(1317)+MKI$(8191)+MKI$
   (2634)+MKI$(8190)+MKI$(3220)+MKI$(16380)+MKI$(5416)
9  a$=a$+MKI$(16376)+MKI$(4688)+MKI$(32752)+MKI$(8608)+MKI$(32736)
   +MKI$(8384)+MKI$(65472)+MKI$(21248)+MKI$(65280)+MKI$(19456)+
   MKI$(64512)+MKI$(61440)+MKI$(61440)+MKI$(49152)
10 PELLIPSE 166,24*8-4,74,8
11 FOR i=0 TO 16*PI STEP 0.05
12 SETCOLOR 15*(-(i=2*PI)-(i MOD 2*PI<0.1)),-15*(i=2*PI)-INT(RND
   *15)*(i MOD 2*PI<0.1),-15*(i=2*PI)-INT(RND*15)*(i MOD
   2*PI<0.1),-15*(i=2*PI)-INT(RND*15)*(i MOD 2*PI<0.1)
13 SPRITE a$,COS(i)^(4*INT(i/8/PI)+1)*80+160,SIN(i)^(4*((INT
   (i/(2*PI))) MOD 4)+1)*80+95
14 PLOT COS(i)^(4*INT(i/8/PI)+1)*80+160,SIN(i)^(4*((INT(i/(2*PI)))
   MOD 4)+1)*80+95
15 PRINT AT(15,24);MID$("              SALUT A TOUS !!!!!   F L A
   G N U M    VOUS PRESENTE LA 'PARAMETRIC DEMO'    ( non qu'elle
   soit paramétrable -rassurez vous- mais le dessin là haut ^
   ,c'est une courbe paramétrée.",(INT(i*10)) MOD 190,-
   14*(i<6*PI))
16 PRINT AT(15,24);MID$("              UN PETIT BONJOUR A 1024 (LE
   BAC APPROCHE HEHE...)      ET  A  ZOCH     (MERCI POUR LA
   REPARATION DE DISQUETTE ... A  BIENTOT POUR D'AUTRES PUNCHS
   .....     S A L U T      ",(INT(i*10)) MOD 190,-14*(i=>6*PI))
17 SOUND 1,12,4,INT(RND*12),0.0001
18 PUT 80+COS(i-PI/2)*45,1,p$
19 PLOT COS(i*15)*5*INT(i/2/PI)-20*(i MOD 2*PI<=PI)-300*(i MOD
   2*PI>=PI),-((i MOD 2*PI)*100)*(i MOD 2*PI<=PI)-((i MOD 2*PI)-
   PI)*100*(i MOD 2*PI>PI)
20 NEXT i
```

Comme à la parade, Flagnum continue avec un jeu de robots à tomber par terre, on en redemande, bravissimo! Au joystick.

```
1  DIM t(40,40),m(4),ms(4)
2  FOR i=0 TO 919
3  t(-(i MOD 40)*((i MOD 2)=0)+(i MOD 40)*(RND<0.6)*(i MOD 2<>0),-
   INT(i/40)*(INT(i/40) MOD 2=0)+INT(i/40)*(RND<0.6)*(INT(i/40)
   MOD 2<>0))=1
4  t(i MOD 40,INT(i/40))=-(t(MAX((i MOD 40)-1,0),MAX(INT(i/40)-
   1,0))<>1)*t(i MOD 40,INT(i/40))
5  t(i MOD 40,INT(i/40))=((i MOD 40=0)+(INT(i/40)=0)+(i MOD
   40=39)+(INT(i/40)=22)<>0)*(t(i MOD 40,INT(i/40))<>1)-(t(i MOD
   40,INT(i/40))<>0)
6  t(i MOD 40,INT(i/40))=(t(i MOD 40,INT(i/40))=0)*(RND<0.05)*2
   +t(i MOD 40,INT(i/40))
7  PRINT AT(-(i MOD 40)*(t(i MOD 40,INT(i/40))>0)+1,-INT(i/40)*
   (t(i MOD 40,INT(i/40))>0)+1);MID$("#"+CHR$(127),-(t(i MOD 40,
   INT(i/40))=1)-2*(t(i MOD 40,INT(i/40))=2),1)
8  NEXT i
9  FOR i=0 TO 25000
10 u=i MOD 4
11 PRINT AT(-INT(s/100)*(as<>0)+1-(as=0),-(s MOD 100)*(as<>0)+1-
   23*(as=0));" ";AT(18,24);LEFT$(CHR$(7),-(t(INT((s+as)/100),
   (s+as) MOD 100)=2));"score :";sc;"      ";AT(-INT(m(u)/100)*(ms
   (u)<>0)+1,-(m(u) MOD 100)*(ms(u)<>0)+1);" ";
12 s=s+as-(i=0)*1111
13 m(u)=m(u)+ms(u)-(i=0)*203-(i=1)*221-(i=2)*3802-(i=3)*3821
14 as=(((STICK(1)=4)*(t(INT(s/100)-1,s MOD 100)<>1)*(INT(s/100)>1)
   -(STICK(1)=8)*(t(INT(s/100)+1,s MOD 100)<>1)*(INT(s/100)<=40)))
   *100+(STICK(1)=1)*(t(INT(s/100),(s MOD 100)-1)<>1)*(s MOD 100>
   1)-(STICK(1)=2)*(t(INT(s/100),(s MOD 100)+1)<>1)*(s MOD 100<30)
15 ms(u)=SGN(m(u)-s)*(INT(m(u)/100)>1)*(m(u) MOD 100<40)*(t(INT(m
   (u)/100)-SGN(m(u)-s),m(u) MOD 100)=0)*100+SGN((m(u) MOD 100)-(s
   MOD 100))*(m(u) MOD 100<23)*(m(u) MOD 100>=1)*(t(INT(m(u)/100),
   (m(u) MOD 100)-SGN((m(u) MOD 100)-(s MOD 100)))=0)
16 sc=sc-(t(INT(s/100),s MOD 100)=2)
17 t(INT(s/100),s MOD 100)=0
18 PRINT AT(INT(s/100)+1,(s MOD 100)+1);"o";AT(INT(m(u)/100)+1,
   (m(u) MOD 100)+1);"+";
19   EXIT IF m(u)=s
20 NEXT i
```



Arnauld Chevallier aime bien les démos, en voici une nouvelle avec de la musique s'il vous plait!

```
1  t$="EELJEELJCCJHCCJHEEJLEEJLCCJHCCJHLECEJECEHECEHJLQQQ SALUT A
   TOUS !!      VOICI UNE DEMO PUNCH, AVEC UN LOGO PUNCH, UN
   SCROLLING PUNCH, ET UNE MUSIQUE PUNCH !! ESSAYEZ D'APPUYEZ SUR
   LES 'Shift' OU SUR 'Control' ..."
2  DEFFILL 2
3  PRBOX 0*XBIOS(7,0,772)*XBIOS(7,15,1888)*XBIOS(7,1,0)*XBIOS(7,
   14,1655),110,319,140
4  GRAPHMODE 3
5  PBOX 0*BIOS(11,0),120,319,130
6  DEFTEXT 1,16,0,12
7  FOR a%=0 TO 1
8  TEXT 3+a%,131+a%,315,"VIVE LES GFA-PUNCHS"
9  TEXT 6+a%,180+a%,"~~~* PUNCH *~~~"
10 NEXT a%
11 FOR l%=0 TO 10000
12 BMOVE XBIOS(2)+4640,XBIOS(2)+4640+160*INT(0.5-SIN(l%/25*PI)*5),
   18080
13 IF l% MOD 4=0
14 PRINT AT(39,1);MID$(t$,51+(l%/4) MOD 180,1)
15 ENDIF
16 GET 2+BIOS(11,-1),0,319,7,s$
17 PUT 0,0,s$
18 VSYNC
19 SOUND 1,15-(l% MOD 10)/2,ASC(MID$(t$,l%/5 MOD 50+1,1))-64,3+l%
   MOD (1+2*(INT(l%/250) MOD 2))
20 NEXT l%
```

Cyril Guerin nous propose le scrolling de deux images Degas, un effet sympathique.

```
1  PRINT "Selectionner les deux images. La première d'abord"
2  FILESELECT "a:\*.pil",a$,nom$
3  FILESELECT "a:\*.pil",a$,nom2$
4  BLOAD nom$,XBIOS(2)-34+XBIOS(5,L:1015808-32000,L:-1,W:0)*0
5  BLOAD nom2$,XBIOS(3)-34+XBIOS(6,L:XBIOS(2)-32)*0
6  BMOVE XBIOS(2),1015808-64034,32000
7  FOR y|=199 TO 0 STEP -1
8  RC_COPY 1015808-64034,0,-(y|-99)*(y|>100),320,200+100*(y|<=100)
   -y| TO 1015808-64034,0,-(y|-100)*(y|>100)-1*(y|<=100)
9  RC_COPY XBIOS(3),0,y|-99*(y|<=100),320,1 TO 1015808-64034,0,-
   (99)*(y|>100)
10 IF y|<=100
11 RC_COPY 1015808-64034,0,103-y|,320,y|-3 TO 1015808-64034,0,102-y|
12 RC_COPY XBIOS(3),0,y|,320,1 TO 1015808-64034,0,99
13 ENDIF
14 BMOVE 1015808-64034,XBIOS(2),32000
15 NEXT y|
16 FOR y|=0 TO 99
17 RC_COPY 1015808-64034,0,0,320,100+y| TO 1015808-64034,0,1
18 RC_COPY XBIOS(3),0,99-y|,320,1 TO 1015808-64034,0,0
19 BMOVE 1015808-64034,XBIOS(2),32000+XBIOS(6,L:1015808-32032)*0
20 NEXT y|
```

Un petit jeu en GfA proposé par Flagnum, ce n'est pas un Punch, il faut sortir du labyrinthe.

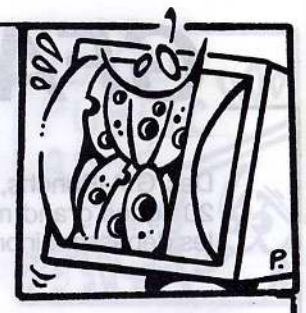
Les commandes:

8 pour avancer — 6 pour aller à droite
4 pour aller à gauche — Return pour le plan

```
+ DIM t(40,40)
+ x=11
+ y=21
+ FOR i=0 TO 919
+ t(-(i MOD 40)*((i MOD 2)=0)+(i MOD 40)*(RND<0.6)*(i MOD 2<>0),-
  INT(i/40)*(INT(i/40) MOD 2=0)+INT(i/40)*(RND<0.6)*(INT(i/40)
  MOD 2<>0))=1
+ t(i MOD 40,INT(i/40))=-(t(MAX((i MOD 40)-1,0),MAX(INT(i/40)-
  1,0))<>1)*t(i MOD 40,INT(i/40))
+ t(i MOD 40,INT(i/40))=((i MOD 40=0)+(INT(i/40)=0)+(i MOD
  40=39)+(INT(i/40)=22)<>0)*(t(i MOD 40,INT(i/40))<>1)-(t(i MOD
  40,INT(i/40))<>0)*(i<>900)*(i<>860)
+ PBOX -(i MOD 40)*8*(t(i MOD 40,INT(i/40))=1),-INT(i/40)*8*(t(i
  MOD 40,INT(i/40))=1),-(i MOD 40)*8*(t((i MOD
  40),INT(i/40))=1)+7,-INT(i/40)*8*(t((i MOD 40),INT(i/40))=1)+7
+ NEXT i
+ SGET laby$
+ REPEAT
+ a$=INPUT$(1)
+ d=d+(a$="4")-(a$="6")+25000
+ l=ABS(d MOD 4)
+ IF a$=CHR$(13)
+ SPUT laby$
+ PRINT AT(x+1,y+1);MID$("^>v<",l+1,1)
+ VOID INP(2)
+ ENDIF
+ x=x-(ABS(d MOD 4)=1)*(t(x+1,y)=0)*(a$="8")+(ABS(d MOD
  4)=3)*(t(x-1,y)=0)*(a$="8")
+ y=y-(ABS(d MOD 4)=2)*(t(x,y+1)=0)*(a$="8")+(ABS(d MOD
  4)=0)*(t(x,y-1)=0)*(a$="8")
+ CLS
+ FOR i=3 TO 15
+ u=(COS(i*PI/180))^180
+ uk=COS((i-1)*PI/180)^180
+ a=x-(i-3)*((l=1)-(l=3))
+ b=y-(i-3)*((l=2)-(l=0))
+ EXIT IF t(a,b)=1
+ DRAW 160-(29-i/3)*6*u,100-(29-i/3)*3.5*u TO 160-(29-
  i/3)*6*u,100+(29-i/3)*3.5*u TO 160+(29-i/3)*6*u,100+(29-
  i/3)*3.5*u TO 160+(29-i/3)*6*u,100-(29-i/3)*3.5*u TO 160+(29-
  i/3)*6*u,100
+ LINE 160+(29-(i-1)/3)*6*uk,100-(29-(i-1)/3)*3.5*uk*(t(a-
  (l=0)+(l=2),b-(l=1)+(l=3))=1)-(29-i/3)*3.5*u*(t(a-
  (l=0)+(l=2),b-(l=1)+(l=3))=0),160+(29-i/3)*6*u,100+(29-
  i/3)*3.5*u
+ LINE 160+(29-(i-1)/3)*6*uk,100+(29-(i-1)/3)*3.5*uk*(t(a-
  (l=0)+(l=2),b-(l=1)+(l=3))=1)+(29-i/3)*3.5*u*(t(a-
  (l=0)+(l=2),b-(l=1)+(l=3))=0),160+(29-i/3)*6*u,100-(29-
  i/3)*3.5*u
+ LINE 160-(29-(i-1)/3)*6*uk,100-(29-(i-1)/3)*3.5*uk*(t(a+(l=0)-
  (l=2),b+(l=1)-(l=3))=1)-(29-i/3)*3.5*u*(t(a+(l=0)-
  (l=2),b+(l=1)-(l=3))=0),160-(29-i/3)*6*u,100+(29-i/3)*3.5*u
+ LINE 160-(29-(i-1)/3)*6*uk,100+(29-(i-1)/3)*3.5*uk*(t(a+(l=0)-
  (l=2),b+(l=1)-(l=3))=1)+(29-i/3)*3.5*u*(t(a+(l=0)-
  (l=2),b+(l=1)-(l=3))=0),160-(29-i/3)*6*u,100-(29-i/3)*3.5*u
+ NEXT i
+ BOX 160-(29-(i-1)/3)*6*uk,100-(29-(i-1)/3)*3.5*uk,160+(29-(i-
  1)/3)*6*uk,100+(29-(i-1)/3)*3.5*uk
+ UNTIL y>=22
```

GFA-PUNCHS

Un GfA-Punch, c'est plein de couleurs, de sons, d'algorithmes compliqués, de formules mathématiques, ça devrait être interdit...



Vous savez jouer à Tron? Trouvez-vous un copain, deux joysticks et tapez ce GfA-Punch de Flagnum.

```
1  FOR i=0 TO 179
2  BOX 16*(i MOD 18)+16,INT(i/18)*16+16,16*(i MOD 18)+32,INT(i/18)
   *16+32
3  NEXT i
4  REPEAT
5  tt1=-(POINT(x1+((INT(u/100)=4)-(INT(u/100)=8))*16+8,y1+((INT(u
   /100)=1)-(INT(u/100)=2))*16+8)=0)
6  tt2=-(POINT(x2+((u MOD 100=4)-(u MOD 100=8))*16+8,y2+((u MOD
   100=1)-(u MOD 100=2))*16+8)=0)
7  PBOX -x1*(tt1<>0),-y1*(tt1<>0),-(x1-16*(x1<>0))*(tt1<>0),-(y1-
   16*(x1<>0))*(tt1<>0)
8  PBOX -x2*(tt2<>0),-y2*(tt2<>0),-(x2-16*(x2<>0))*(tt2<>0),-(y2-
   16*(y2<>0))*(tt2<>0)
9  y1=y1+((-(y1>16)*(INT(u/100)=1)+(INT(u/100)=2)*(y1<152))*16)
   *tt1-16*(x1=0)
10 x1=x1+((x1<288)*(INT(u/100)=8)-(INT(u/100)=4)*(x1>16))*16*tt1-
   16*(x1=0)
11 y2=y2+((-(y2>16)*(u MOD 100=1)+(u MOD 100=2)*(y2<152))*16)*tt2-
   16*10*(y2=0)
12 x2=x2+((x2<288)*(u MOD 100=8)-(u MOD 100=4)*(x2>16))*16*tt2-
   16*18*(x2=0)
13 score=score-(y1>16)*(y1<152)*(x1<288)*(x1>0)*(tt1=1)*100-(tt2
   =1)*(y2>16)*(x2>0)*(y2<152)*(x2<288)
14 PBOX x1,y1,x1+16,y1+16
15 PBOX x2,y2,x2+16,y2+16
16 PRINT AT(1,24);"  joueur 1 : ";INT(score/100);"          joueur
   2 : ";(score MOD 100);CHR$(-7*(tt1<>0 OR tt2<>0))
17 u=STICK(1)*100+STICK(0)
18 UNTIL (POINT(x1+8,y1-8)<>0 AND POINT(x1-8,y1+8)<>0 AND POINT(x1
   +24,y1+8)<>0 AND POINT(x1+8,y1+24)<>0) OR (POINT(x2+8,y2-8)<>0
   AND POINT(x2-8,y2+8)<>0 AND POINT(x2+24,y2+8)<>0 AND POINT(x2
   +8,y2+24)<>0)
19 PRINT AT(1,1);LEFT$(" G A G N A N T          :          JOUEUR     ",-
   36*(score MOD 100<>INT(score/100)));-2*(score MOD
   100>INT(score/100))-(INT(score/100)>score MOD 100);LEFT$("
   M A T C H    N U L",-40*(score MOD 100=INT(score/100)))
20 STICK (0)
```

Denis Ollier, un habitué des GfA-Punchs, nous a préparé une danse des guerriers papous.

```
1  sprl$=MKI$(0)+MKI$(1)+MKI$(1)+MKI$(0)+MKI$(4)+MKI$(960)+MKI$
   (960)+MKI$(2016)+MKI$(2016)+MKI$(26016)+MKI$(26016)+MKI$(26592)
   +MKI$(26592)+MKI$(25152)+MKI$(25152)+MKI$(32766)+MKI$(32766)+
   MKI$(32766)+MKI$(32766)+MKI$(2022)+MKI$(2022)+MKI$(2022)+MKI$
   (2022)
2  sprl$=sprl$+MKI$(16358)+MKI$(16358)+MKI$(16352)+MKI$(16352)+
   MKI$(12384)+MKI$(12384)+MKI$(61536)+MKI$(61536)+MKI$(61536)+
   MKI$(61536)+MKI$(120)+MKI$(120)+MKI$(120)+MKI$(120)
3  spr2$=MKI$(0)+MKI$(1)+MKI$(1)+MKI$(0)+MKI$(4)+MKI$(960)+MKI$
   (960)+MKI$(2016)+MKI$(2016)+MKI$(1446)+MKI$(1446)+MKI$(2022)+
   MKI$(2022)+MKI$(582)+MKI$(582)+MKI$(32766)+MKI$(32766)+MKI$
   (32766)+MKI$(32766)+MKI$(26592)+MKI$(26592)+MKI$(26592)+MKI$
   (26592)
4  spr2$=spr2$+MKI$(26620)+MKI$(26620)+MKI$(2044)+MKI$(2044)+MKI$
   (1548)+MKI$(1548)+MKI$(1551)+MKI$(1551)+MKI$(1551)+MKI$(1551)+
   MKI$(7680)+MKI$(7680)+MKI$(7680)+MKI$(7680)
5  a$=MKI$(&H20)+MKI$(&H750)+MKI$(&H7)+MKI$(&H7)+MKI$(&H700)
6  ~XBIOS(6,L:VARPTR(a$))
7  SPRITE sprl$,0,0
8  SPRITE spr2$,20,0
9  SGET e$
10 CLS
11 s&=20
12 EVERY 50 GOSUB a_que_anime
13 REPEAT
14   RC_COPY V:e$,s&,0,16,16 TO XBIOS(2),ADD(55,50*SINQ(b-20)),2
15   ADD b,ABS(2*COSQ(2))
16   BMOVE XBIOS(2),ADD(XBIOS(2),364),32000
17 UNTIL MOUSEK
18 PROCEDURE a_que_anime
19   SWAP s&,s2&
20 RETURN
```

Tous ces Punchs, à l'exception d'un seul, tournent en Basse Résolution.

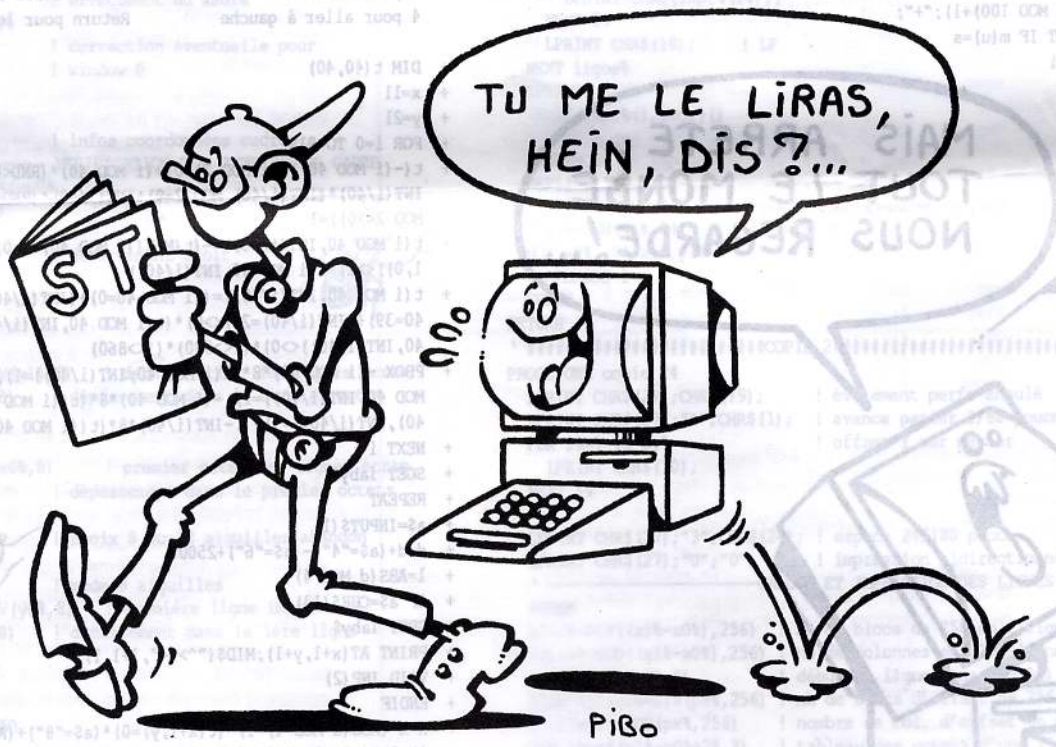
Ils sont téléchargeables sur le 3615 STMAG.

Arnauld Chevallier déplace un sigle en fil de fer, c'est surprenant.

```
1  DATA 4,0200101101,3,00021210,3,02001210,3,10000212,5,000201111012
2  e$=STRING$(32256,0)
3  VOID XBIOS(5,L:INT((VARPTR(e$)+256)/256)*256,L:-1,W:-1)+
   XBIOS(7,0,3)+XBIOS(7,15,&H770)
4  FOR k=0 TO 100 STEP 0.1
5  VOID XBIOS(5,L:XBIOS(2),L:XBIOS(3),W:-1)
6  CLS
7  BOX 100+COS(k)*100,100+COS(k)*100,220-COS(k)*100,100-COS(k)*100
8  FOR a%=0 TO 4
9  READ np%,a$
10 FOR b%=0 TO np%
11 PLOT x%,y%
12 x%=160+(VAL(MID$(a$,b%*2+1,1))+a%*2-5)*40*COS(k*1.1)
13 y%=100+(VAL(MID$(a$,b%*2+2,1))-1)*50*SIN(k)
14 IF b%
15 DRAW  TO x%,y%
16 ENDIF
17 NEXT b%
18 NEXT a%
19 RESTORE
20 NEXT k
```

Denis Ollier nous offre ici son scrolling différentiel. Il faut obligatoirement être équipé d'un blitter.

```
1  DRAW 0,120 TO 110,88 TO 180,120 TO 250,90 TO 319,120
2  DRAW 0,190 TO 50,160 TO 120,180 TO 180,155 TO 240,195 TO
   319,190
3  CIRCLE 80,40,19
4  FOR i|=0 TO 3
5    READ c%,x%,y%
6    DEFFILL c%,1  !Palette systeme.
7    FILL x%,y%
8  NEXT i|
9  SGET ecr$
10 adr%=V:ecr$
11 REPEAT
12   RC_COPY adr%,0,88,1,66 TO adr%,319,88 !BLITTER OBLIGATOIRE.
13   RC_COPY adr%,1,88,320,66 TO adr%,0,88
14   RC_COPY adr%,0,154,4,46 TO adr%,316,154
15   RC_COPY adr%,3,154,316,46 TO adr%,0,154
16   VSYNC
17   BMOVE ADD(adr%,13920),ADD(XBIOS(2),13920),18000
18 UNTIL MOUSEK
19 DATA 5,300,20,6,80,40,9,150,140,8,150,195
```



Flagnum est amoureux et il le montre.

```
1  FOR n%=0 TO 7
2  SETCOLOR 7-n%,7-n%,7-n%,7
3  SETCOLOR n%+8,7,7-n%,7-n%
4  NEXT n%
5  SETCOLOR 0,0,0,0
6  ATEXT 10,90,0,"LE PARCOURS DU "
7  FOR g=0 TO 0.13 STEP 0.01
8  FOR a=0 TO 2*PI STEP 0.01
9  y%=120-COS(a)*40-SQR(ABS(x%))*10
10 x%=SIN(a)*100
11 PSET COS(g)*(x%+170)-SIN(g)*y%,SIN(g)*(x%+170)+COS(g)*y%,
   SQR(x%*x%+y%*y%)/10
12 NEXT a
13 NEXT g
14 FOR i=-PI/2 TO PI STEP 0.01
15 PSET SIN(i)*25*i,-COS(i)*25*i+110,SQR((SIN(i)*25*i)^2+(110-
   COS(i)*25*i)^2)/10+7
16 PSET -(COS(i+3.6)*50+110)*(i<PI/8-0.6),-(SIN(i+3.6)*100+100)*
   (i<PI/8-0.6),-(SQR((SIN(i)*25*i)^2+(110-COS(i)*25*i)^2)/10)*
   (i<PI/8-0.6)
17 PSET -(COS(i+3.6)*50+111)*(i<PI/8-0.6),-(SIN(i+3.6)*100+100)*
   (i<PI/8-0.6),-(SQR((SIN(i)*25*i)^2+(110-COS(i)*25*i)^2)/10)*
   (i<PI/8-0.6)
18 PSET SIN(i)*25*i+1,-COS(i)*25*i+110,SQR((SIN(i)*25*i)^2+(110-
   COS(i)*25*i)^2)/10+7
19 NEXT i
20 VOID INP(2)
```

C'est le seul Punch qui tourne en monochrome, il est de Stéphane Bédon, c'est un jeu assez simple

```
1  DIM ecr$(24)
2  TIME$="00:00:00"
3  space_hunter(XBIOS(2)+2560,XBIOS(2)+1280,40,3,3)
4  PROCEDURE space_hunter(ecr1%,ecr2%,x|,y|,vies|)
5  REPEAT
6  k$=UPPER$(INKEY$)
7  ADD x|,SUB(k$="W" AND x|>1,k$="X" AND x|<80)
8  ADD y|,SUB(k$="M" AND y|>3,k$=":" AND y|<23)
9  MID$(ecr$(24),RANDOM(79))="**"
10 MID$(ecr$(24),RANDOM(80))=LEFT$(CHR$(249),-(RANDOM(5)>2))+LEFT$
   ("x",-(RANDOM(5)=4))
11 PRINT AT(x|,y|);"V"
12 PRINT AT(1,1);"Score: ";score&;TAB(19);"Koduks: ";koduks&;TAB
   (43);"Temps: ";TIME$;TAB(71);"Vies: ";vies|;AT(1,24);ecr$(24);
13 DELETE ecr$(2)
14 ecr$(24)=SPACE$(79)
15 SUB score&,MUL(10,MID$(ecr$(y|),x|,1)=CHR$(249))+MUL(100,MID$
   (ecr$(y|),x|,1)="x")
16 ADD vies|,MID$(ecr$(y|),x|,1)="*"
17 BMOVE ecr1%,ecr2%,29440
18 INC koduks&
19 UNTIL vies|=0 OR TIME$>="00:01:30" OR k$=CHR$(27)
20 RETURN
```

Encore un nouveau Punchiste, il s'appelle Arnauld Chevallier, et sa démo ne passe pas inaperçue, bravo pour les fontes!

```
1  DIM 1$(26)
2  c$=MKL$(1111)+MKI$(839)+MKI$(1360)+MKI$(816)+MKI$(1632)+MKI$(10
   88)+MKI$(1904)+STRING$(16,0)
3  FOR a%=0*XBIOS(6,L:VARPTR(c$)) TO 26
   ' remplacez les points par des espaces (contrainte PAO)
4  PRINT AT(1,1);CHR$(a%+64+33*(a%=0));AT(9,a%/2+10);MID$("@@@..@.
   .@.@..@..@@@.@..@@..@.@..@.@@.@.@@...@..@@@.@.@..@.@@.@.@....@@
   @@@@@..@..@.@.@@.@....@..@@....@@.@.@.@@.@@...@..@@....@@@..@..
   @..@@@.@..@",1+INT(a%/4)*24,24)
5  FOR x%=0 TO 48
6  y%=x% MOD 7
7  DEFFILL y%+2
8  PRBOX x%/1.8-COS(y%)*3,y%*4+10+50*(POINT(x%/7,y%)=0),x%/1.8+7,
   y%*4+17+50*(POINT(x%/7,y%)=0)
9  NEXT x%
10 GET 0,10,33,41,l$(a%)
11 PUT 0,10,l$(0)
12 NEXT a%
13 m$=" HI GUYS        I JUST WANNA SHOW YOU THIS PUNCH DEMO WITH
   ITS FURIOUS SCROLLING WHICH COMES FROM MY CRAZY MIND        I HOPE
   IT ISNT TOO FAST FOR YOU      BE CAREFUL NOT TO BECOME
   NUTS"+SPACE$(20)
14 FOR z%=0 TO 10000
15 k%=(k%+1) MOD LEN(m$)*2
16 BMOVE XBIOS(2)+8,XBIOS(2),5120
17 PUT 304+16*ODD(k%),0*XBIOS(7,15,(7-INT(z%/10) MOD 8)*VAL(MID$
   ("0012572562720160172 73",1+(INT(z%/80) MOD 7)*3,3))),l$(ASC
   (MID$(m$,k%/2,1))-64-32*(MID$(m$,k%/2,1)=" "))
18   VSYNC
19   SOUND 1,15,#z%^2
20 NEXT z%
```

Formation à l'assembleur pratique

Formation au PROCESSEUR 88888

UN BREF APERÇU HISTORIQUE

1980: le 4040
1991: le 88888

FAISONS LA LUMIERE SUR LE LANTERNEAU DES FONDEURS DE SILICIUM

(Note Au Claviste: j'ai bien écrit "lumière", il ne s'agit pas d'une erreur, c'est un mot d'auteur. Je connais les expressions de la langue française non mais).

Au début des années 90, les fondeurs (frondeurs? ah, ah, ah.) rivalisaient d'imagination pour imaginer un nouveau concept non imaginé encore par des confrères pourtant imaginatifs et qui allaient leur assurer la puissance et la gloire dans cet univers pitoyable.

Ainsi a-t-on vu apparaître les processeurs CISC (Complex Instruction Set Computer? euh Coprocessor? euh Catéchisme?).
Autrement dit, PIB, Processeur à Instructions Bordéliques, et non Produit Intérieur Brut, comme les économistes veulent nous le faire croire, c'est pour ça qu'ils se plantent toujours.

Certains esprits avisés firent remarquer:

- Que finalement, statistiquement, peu d'instructions du processeur sont utilisées, donc autant ne programmer sur le processeur que celles-ci.
- Que si on ne programme (en fait, dans ce cas, on "câble") que peu d'instructions, on pourra les faire plus rapides, voire même s'exécuter en un seul cycle d'horloge, ce qui ne fait pas 24 heures et encore moins 12 heures.
- Que les compilateurs ne sont de toutes façons pas capables de tirer parti de centaines d'instructions différentes, mordorées et criardes.
- Que le Complex faisait plus sûrement référence au fait que la programmation était complexe (interprétation personnelle, pardon de me placer momentanément dans les esprits avisés, mais si je la fais plus tard, elle sera hors contexte, donc je la fais maintenant, ne m'en veuillez pas. Rassurez-vous, je ne suis pas, mais alors, pas du tout, avisé, sauf parfois dans la rue, un quidam m'avise et me demande l'heure, et je lui réponds alors "Michel, t'as qu'à acheter une montre, tu es toujours en retard dans la programmation de tes serveurs.". Ce à quoi vous rétorquerez, cher lecteur - il s'agit bien entendu d'un vous de politesse, car je sais très bien qu'à ce stade du cours - ah, d'habitude, on court dans un stade, ah - il ne reste plus que Mic Dax, de son vrai nom Pericolo e Miniteli, pour me suivre - que s'il se nomme Michel, DONC, ce n'est pas tout à fait un quidam. Enfin, le fait que je connaisse son nom laisse supposer que je le connais. Ce peut donc être un quidam pour ceux qui ne le connaissent pas, mais pour moi ce n'en est pas un. D'ailleurs, on peut généraliser à tous les prénoms, ça marche aussi si ce n'est pas Michel, essayez, vous verrez. C'est fou comme une simple phrase comme "Un quidam m'avise." peut faire réaliser à quel point notre vision du monde dépend du point de vue, et la focalo, et tant va la cruche à l'eau qu'à la fin elle se mouille.).

DONC les fondeurs inventèrent des processeurs d'une nouvelle génération, ce qui fait une liaison avec le chapitre suivant.
Avez-vous remarqué comme le fait de mettre "donc" vous fait croire que le chapitre suivant est introduit d'une manière logiquement irréfutable? Ce n'est qu'illusion, méfiez-vous des escrocs qui mettent des "donc" pour cautionner une argumentation par trop légère. C'est mon cas.

INTRODUCTION DES PROCESSEURS A (HAUT) RISC

Alors là, tu me dis, Mic - tu permets que je t'appelle au téléphone? - que celle-là elle est éhontement éculée.
Et alors là je te réponds: oui.

Donc, les frondeurs pompèrent les études faites à l'Université de Berkeley, mère de toutes les innovations et commençûrent à mettre sur le marché des processeurs dits "RISC".
RISC, c'est Reduced Instruction Set Computer? euh Coprocessor? euh Catastrophe?. Autrement dit PIRE (Processeur à Instruction Réduites mais Efficaces).
Le concept est le suivant: un nombre réduit d'instructions dans le processeur, mais des instructions efficaces.

C'est de la paraphrase? Oui.

En fait, les processeurs RISC ont un nombre d'instructions très réduit, en général 32 ou 64, ce qui leur permet d'être toutes câblées. Et donc de s'exécuter en un cycle d'horloge, sauf pour quelques-unes genre la multiplication, mais bon, on se démerde dans ce cas avec une architecture dite Pipe-Line-Renaud - et non pas Line pipe euh bon bref - pour faire croire que c'est en une seule instruction. Bon j'arrête, je commence à vraiment expliquer, ce n'est pas le but. (Mic: - si. Moi: - m'en fous.)

DONC nous en venons au 88888.

LE 88888 de FYC

La sortie du 88888 est due à la célèbre shoot'em up américaine FYC (Fuck Your Computer), filiale de la non moins non-confidentielle société française NTM.

Cette société a poussé à l'extrême les recherches sur les processeurs RISC et a inventé le OOOC (Only One Operation Computer? euh Coprocessor? Euh..?).

OOOC, ça veut dire processeur à une seule instruction, je n'ai aucune abréviation marrante à vous proposer, pardon.

Ce processeur, dont la complexité interne n'a d'égale que sa facilité d'utilisation, donc, ne propose qu'une seule instruction, le NOP.

APARTE DISCRET

Là, je vous raconte une anecdote pour prouver que je m'intéresse aux potins de la profession: il se trouve que FYC vient de rompre sans ménagement avec le fameux Green Hills qui développait le compilateur C et Fortranvan 77 - Fortran 77 a été renommé depuis le rachat de tous les droits par Sony. - car à son sens les compilateurs ne tiraient pas parti de toutes les possibilités du processeur.
A mon avis, ce sont des histoires de gros sous, ils disent ça pour amuser la galerie, c'est magouilles et compagnie, on me la fait pas à moi.

RECHERCHE FONDAMENTALE

DONC, je précise que la recherche fondamentale effectuée par NTM et reprise à son compte par FYC a abouti à une technologie révolutionnaire, non encore imaginée par les concurrents, et prenant à contre-pied les-dits concurrents non aussi imaginatifs dans la recherche fondamentale, là où pourtant les cerveaux les plus imaginatifs rivalisent d'imagination, enfin j'imagine.

Pendant que le monde entier s'est engagé dans une course effrénée à la vitesse d'horloge, FYC avait décidé d'échantillonner sa puce à 2Hz, les versions livrées aux constructeurs seront finalement à 1Hz, et une version à 0,125Hz est déjà dans les cartons.

Avantage déterminant: l'opérateur a le temps de recopier en temps réel les ordres sur son Atari (la version à 2Hz exige une frappe rapide, mais la version 0,125Hz permettra l'embauche de main d'oeuvre sous-qualifiée, FYC résout de ce fait un des plus grands dilemnes de cette fin de siècle: si j'informatise pas je meurs, si j'informatise je mets au chômage les travailleurs sous-qualifiés).

CA, C'EST LACHE!

Quel dommage! Le nombre de signes qui m'était alloué pour ce cours d'assembleur 88888 est atteint! Je ne pourrai donc pas vous parler plus avant des nombreuses optimisations de ce processeur, pas plus que je ne pourrai commenter le listing exemple - une simulation de Plextel - que j'avais préparé à votre attention.

Mais qu'importe, nous nous retrouverons dans le prochain numéro (n'essayez pas d'acheter plusieurs numéros en kiosque, hein, le contenu est le même, il faut tout leur dire? A la réflexion, si, achetez-en plusieurs, ça fera plaisir au Boss.).

Ah, au fait, je ne le dis pas parce que je suis modeste mais je viens de vous donner un scoop, le 88888 ne sortira qu'en 2001 - au Lycée de l'Espace, centre de recherche semi-conducteur de la NASA -, j'ai juste eu une version de test assez jouable, mais FYC m'a assuré que le version finale comporterait des instructions identiques.
Bon, personne n'est arrivé au bout? J'éteinds le lanterneau et je me casse. Ciao!

Fabrice BROCHE
Alias the bug generator euh the bug killer euh Byte();.

(NAC: c'est bien "lanterneau", hein, pas "landerneau", hein, n'allez pas mettre mon effet par terre, on est déjà si peu de choses.)
(Note Du Claviste A l'Auteur: mmmmhhhh?).

COUPONS LA TETE AUX FAISEURS DE MAUVAIS JEUX DE MOTS !

Formation au RISC / CISC

LES PROCESSEURS RISC SONT-ILS "DÉBILES" ?

Le micro-processeur, comme vous devez le savoir, est la partie la plus importante d'un ordinateur. C'est son intelligence, s'il est possible de parler "d'intelligence". En fait, le micro-processeur est le centre nerveux de la machine, c'est lui qui exécute le programme (ou qui se plante quand il essaie de décoder les embrouillaminis du "Rédacteur") et donne les ordres aux composants associés. Seulement voilà, il y a plusieurs types de processeurs qui s'affrontent, et qui regroupent les utilisateurs en deux "écoles"...

D'un côté, le CISC, ou processeur à jeu d'instructions complexe, a été le premier à voir le jour. Il a été grandement représenté avec les 6502, Z80, 68000/20/30 ou encore 8086/286/386. A l'opposé, le RISC, ou processeur à jeu d'instructions réduit, ne contient donc que peu d'instructions. Sa structure permet alors d'accélérer les traitements, et d'effectuer des anticipations. Mais l'on est en droit de se poser la question suivante : "Mieux vaut-il une tête bien faite ou une tête bien pleine?". Nous nous baserons sur la résolution de deux problèmes pour tenter de répondre à cette question.

Imaginons un instant que l'on soit dans la quatrième dimension, et que Albert Einstein soit un androïde basé sur un processeur CISC. Son ami, "Sylvestre" Stallone, est lui équipé d'un modèle RISC. Albert connaît beaucoup de choses, a un esprit déductif, mais travaille moins vite que Sylvestre. Ce dernier ne sait rien, ou alors tout juste le prénom de sa petite amie (elle aussi androïde), mais est doué d'une rapidité de calcul époustouflante.

"CISC"

"RISC"

Afin de les départager, nous les confrontons à deux problèmes :
- Y a t-il une équivalence entre l'énergie E et la masse m d'un corps, sachant qu'il faut tenir compte de la célérité de lumière ? Si oui, la trouver.
- Quelles sont, parmi les conséquences suivantes, les effets initiaux de l'arme nucléaire sur l'homme : vomissements, fièvres, diarrhées, cécité, brûlures, cyanose, constriction et enfin éclatements d'organes ?

Le premier sujet, Albert, comprend que les questions sont "à tiroir", et qu'il lui faut répondre à la première avant de passer à la seconde. Comme il est relativement lent, il met quelques années à trouver le résultat E=mc2. Puis c'est grâce à cette formule (qui est à l'origine de la libération de l'énergie nucléaire) que ses camarades réalisent la première bombe atomique. Il ne lui reste plus qu'à attendre une expérimentation "grandeur réelle", soit l'attaque d'Hiroshima, pour déterminer les effets exacts de l'arme nucléaire...

Sylvestre, basé sur un processeur RISC, est bien trop rapide. Il brûle la première étape pour expérimenter la seconde. Il arrive à se procurer une bombe atomique, s'enferme chez-lui et actionne le détonateur. Malheureusement, il n'a pas le temps d'en noter les effets, et de toutes façons, il ne sait pas écrire !

Voici donc en quelques lignes la démonstration de la suprématie des micro-processeurs CISC. Alors, si votre revendeur habituel vous propose d'échanger "pour pas cher" votre 68000 contre un 88000, méfiez-vous !

Sébastien MOUGEY
(qui a oublié de mettre des intertitres)

CONCOURS STMAGICIEL SUZUKI : GAGNEZ UNE

Une vraie !

Chez vous !

Maintenant !

En bleu !

Pour vous !

Imaginez la tête des voisins !

... superbe reliure "ST Magazine", d'un beau bleu roi et frappés de notre ancien logo ST Mag. Grâce à notre superbe reliure, ne perdez plus vos numéros, ne les laissez plus prendre la poussière et mettez-les à l'abri des imbéciles. Pour recevoir **CHEZ VOUS** cette superbe reliure, envoyez dès maintenant 5 coupons "POINT CONCOURS" découpés dans STMAGICIEL. Le 100e lecteur nous renvoyant ces coupons recevra **CHEZ LUI** notre superbe reliure. Le concours sera clos le 15 Juillet 1992, date de parution du numéro 2 de STMAGICIEL.

Coupons à renvoyer à STMAGICIEL, 19 rue Hégésippe Moreau, 75018 PARIS. Réglement disponible chez l'Huissier Maître Etmonpoingtuleveuxdanslagueulemonpoingtulevoislàpauvreandouille.

1 POINT CONCOURS

QUE SONT-ILS DEVENUS ?

Les plus attentifs de nos lecteurs auront reconnu dans STMAGICIEL un hommage vibrant à un canard d'informatique ayant sévi entre 1984 et 1986, il s'appelait HEBDOGICIEL et il avait connu un succès phénoménal. Pas fous, nous n'allions pas rendre hommage à un fanzine minable pas connu, ça serait du suicide. On veut bien faire des hommages, à condition que ça rapporte du fric, c'est pas demain la veille qu'on bossera pour des nèfles. Parce que c'est vrai, quoi, HEBDOGICIEL, on s'en tape, mais si ça peut rapporter des millions, on est les premiers à jouer les lèche-culs.
Aujourd'hui encore, pas mal de lecteurs nous écrivent pour savoir ce qu'est devenu HEBDOGICIEL, alors qu'on en sait que d'chi. En fait, si, on sait, mais c'est comme tout, chaque profession a ses petites manies, ses mots tabous. Prenez le théatreux et sa corde, par exemple. HEBDOGICIEL, c'est pareil, mieux vaut ne pas en parler, va savoir pourquoi, encore une connerie de tradition à la con, je sais pas ce qui m'empêche de tout vous dire, tiens.
- psssstttt,
- tiens, salut Godefroy, qu'est-ce que tu veux?
- je peux te parler?
- deux secondes, je mets un truc pour être tranquille.

INTERLUDE

Bon, je vous dirai tout sur la mort d'HEBDOGICIEL dans le numéro 2 de STMAGICIEL, mais alors, tout, tout, tout, ça sera grandiose, il y aura des photos, des déclarations, ne le manquez pas.
En attendant, je peux vous dire ce que sont devenus ceux qui ont titré en 86, à la une d'HEBDOGICIEL, "L'informatique, c'est de la merde"...

Gérard CECCALDI, le Directeur De La Publication, a ouvert un serveur informatique dédié au Mac, il l'a écrit lui-même en Basic. Allez le gonfler sur le 3615 HG1.

Michel DESANGLES, Rédacteur en Chef Adjoint, véritable gourou, est aujourd'hui Rédacteur en Chef Adjoint du magazine Joystick, un canard de micro-informatique, après avoir été Rédacteur en Chef de ST Magazine. Il fait croire à tout le monde qu'il va faire plein d'autres trucs, mais on sait tous que tu n'aimes QUE l'informatique et la presse, Michel, allons.

CARALI, le dessineux, collabore maintenant à Micronews, magazine informatique réalisant une des plus fortes ventes dans ce domaine sur la Bretagne, et continue Le Psikopat, avec Michel, comme quoi je suis qu'une mauvaise langue.

Michaël THEVENET, de la Rédaction (Mich & Micha) écrit à présent pour Univers Mac, un magazine qui traite d'informatique et de Mac...
- psssttt...

INTERLUDE

Pardon: Univers Mac, un EXCELLENT magazine qui traite d'informatique et de Mac.

Stéphane Schreiber, de la Rédaction lui aussi, paie ses dettes de jeu en bossant pour Commodore Amiga Revue, une revue d'informatique.

Sébastien MOUGEY, qui s'occupait de la formation à l'assembleur sur C64, a laissé tomber son micro-serveur ADN pour mieux se consacrer à sa collaboration avec ST Magazine, où il est aujourd'hui Rédacteur en Chef Adjoint Quand Il n'Est Pas En Vacances Ou Au Service Militaire.

Fabrice BROCHE, qui était chargé de la formation à l'assembleur sur Oric, a laissé tomber son Téléstrat, la dope et sa nana pour mieux bosser sur son soft serveur, qu'il a lâchement revendu par derrière à Génération 4 et à Joystick (que des types drôlement fins ont rebaptisés Pénétration 4 et Jastique, c'est fin, hein?).

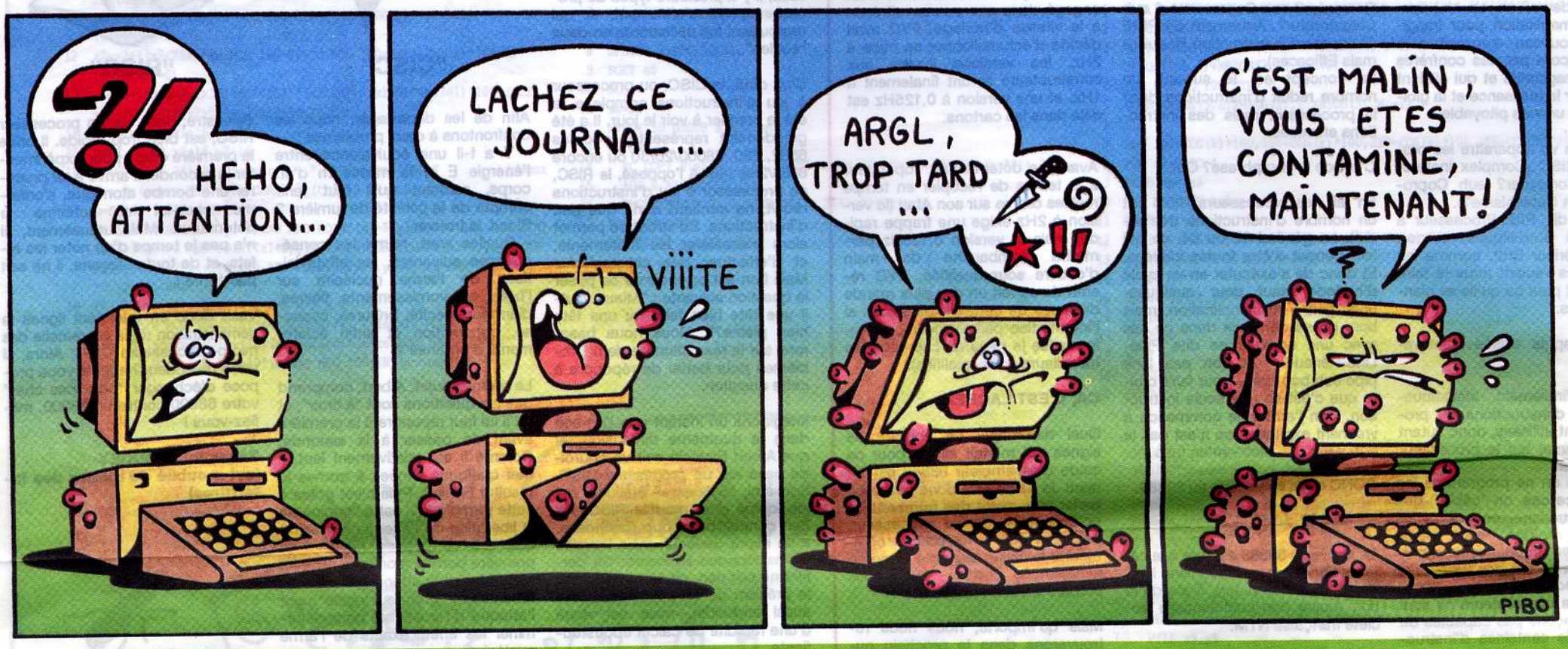
Jean-Marc GASNOT, le Maquettiste, officie actuellement chez Micronews, encore et encore, en tant que Maquettiste, d'accord d'accord.

Les autres sont morts, évanouis dans la nature, ou ont été expulsés pour trouble de l'ordre public, éliminés par le Mossad, on ne sait pas, nous n'avons pas de leurs nouvelles, et on va pas faire des recherches, non plus, rien à battre.

A bientôt pour un prochain numéro de STMAGICIEL, qui prendra certainement un autre nom parce que ça va deux minutes...

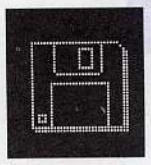
LE MENU

Le menu se trouve dans les pages intérieures, tout bêtement parce que nous sommes sur une page quadri (=quadri-chromie). Les pages en quadri (=couleur) partent en fabrication avant les pages en noir (=noir), il faut le temps de rajouter les couleurs, vous avez 3 jeunes filles vierges qui retouchent la page au pinceau au poil de koala, sinon ça marche pas. L'inconvénient, c'est que ça prend du temps, et qu'il faut prévoir longtemps à l'avance, de préférence à un moment où les pages en noir ne sont pas encore faites. Voilà le hic, le menu n'est pas encore prêt, il reste encore 20 pages en noir à maquetter, et les numéros de page ne sont pas encore posés, c'est un vrai bazar, pas moyen de vous faire un menu. On le fera quand toutes les pages seront faites, sur une page en noir.



PRET À TRAQUER L'INFO A TRAVERS LE MONDE ?

ON SE SENT LE GENIE D'UN CRÉATEUR ?


FATIGUÉ DES PROGRAMMES DES AUTRES ?

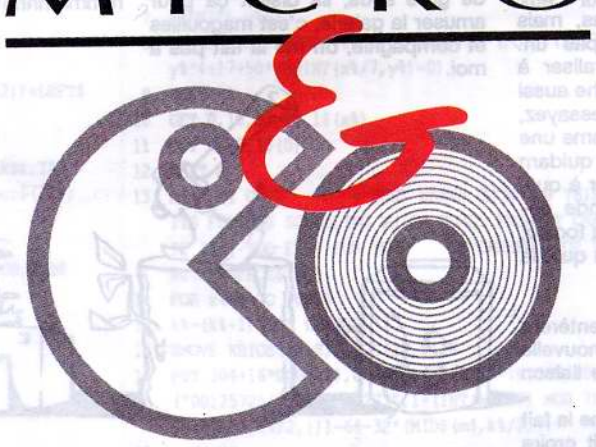
POUR VOUS, L'INFORMATIQUE, C'EST SERIEUX ?

CE QU'IL VOUS FAUT, C'EST UNE PETITE VISITE CHEZ VOTRE KIOSQUIER LE PLUS PROCHE !

ST MAGAZINE, tous les mois en kiosque, 25 francs, l'outil complet pour votre ST.

BUREAUTIQUE ET COMMUNICATION
CRÉATIVITÉ
JEUX VIDEO
FOIRE AUX AFFAIRES
LE VILLAGE CD

MICRO



Du 18 au 21 octobre à la Porte de Versailles

MICRO & Co : 15-17 Avenue Ledru-Rollin 75012 PARIS Tél : 43 44 35 97 Fax : 46 28 89 04